PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

GUÍA DE MEJORES PRÁCTICAS PARA GARANTIZAR SEGURIDAD, INTEGRIDAD Y DISPONIBILIDAD EN BASES DE DATOS

(CASO PRÁCTICO ORACLE 11G Y MYSQL 5.1)

TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS Y COMPUTACIÓN

MARCO VINICIO BURBANO SÁNCHEZ

QUITO 2010

CONTENIDO

RESUMEN

El presente trabajo: Guía de Mejores Prácticas para garantizar Seguridad, Integridad y Disponibilidad en Bases de Datos, busca establecer actividades que permitan una adecuada gestión y administración de cualquier motor de base de datos, ya sean estas de arquitectura comercial o arquitectura libre, mediante la implementación de buenas prácticas que permitan a la organización protegerse de un amplio espectro de amenazas las cuales pueden provenir del interior como del exterior de la organización.

En el Capítulo I se expone una breve introducción a conceptos sobre las Bases de Datos, así como también definiciones sobre Seguridad, Integridad y Disponibilidad. Adicionalmente se emite un enfoque de las mejores prácticas orientadas a la implementación de controles en Bases de Datos.

Para el desarrollo del presente trabajo es necesario el conocimiento de buenas prácticas y el diseño e implementación de bases de datos, que se encuentra documentados en el Capítulo II.

En el Capítulo III, se realiza la implementación de las mejores prácticas definidas, sobre las Bases de Datos seleccionadas. Dando lugar a diferentes formas de obtener el resultado esperado en cada buena práctica dependiendo los objetivos y el motor de base de datos.

En el Capítulo IV se efectúa la comparación de los métodos de configuración y los resultados obtenidos en cada una de las bases de datos.

INTRODUCCIÓN

Desde el inicio de la computación, ya hace más de sesenta años, el hombre ha buscado poder procesar información que le ayude a resolver los problemas de manera mucho más rápida y efectiva.

Posteriormente no bastó con solo procesar la información sino que se encontró con la necesidad de almacenar dicha información con lo cual surgieron los primeros almacenes de información.

A medida que las diferentes industrias y organizaciones crecían la información aumentaba en complejidad y volumen. Dando paso a la necesidad de generar administradores de información los cuales puedan almacenar de manera organizada y permitan la recuperación completa de los datos almacenados.

Por tal razón los controles en el almacenamiento de la información se han convertido en una de las principales prioridades a la hora de establecer una organización, puesto que las compañías para desarrollarse necesitan información y la necesitan rápida, veraz y oportuna**.** Una pequeña falla, alteración o inconsistencia en este vital activo de la empresa puede significar la diferencia entre la excelencia o el fracaso.

La información de la base de datos debe estar protegida contra accesos no autorizados, destrucción o alteración con fines indebidos y la introducción accidental de inconsistencia. Enfrentar cada uno de estos retos implica cubrir necesidades de información mayores día a día.

Como resultado de lo expuesto surge la necesidad de manejar las mejores prácticas a la hora de trabajar con la información, la misma que es de vital importancia en cualquier empresa, pues nadie está dispuesto a salir del mercado por un simple y pequeño error.

El seguimiento de algunos consejos útiles en cuanto a seguridad, integridad y disponibilidad en los almacenes de la información nos permitirán estar día a día en el mercado.

CAPÍTULO 1: DEFINICIONES SOBRE BASE DE DATOS

1.1 DEFINICIONES BÁSICAS

1.1.1 INFORMACIÓN

Podemos definir en primera instancia a la información como un conjunto de datos que se encuentran organizados y que tienen un significado.

La información es el elemento más importante en el proceso de la comunicación, cabe mencionar que es necesario compartir un mismo código entre el que envía y el que recepta la información. Esto no sólo ocurre en un proceso social sino también en el mundo de la informática.

En los últimos años, el avance tecnológico y de la informática devino en que actualmente la información se convierta en un bien muy cotizado por varias personas y/o empresas, todo esto sucedió por la ayuda concedida por la globalización y el aparecimiento de la red más famosa del mundo (Internet), un proceso y un sistema de comunicación que no posee barreras para poder unir un punto del planeta con otro.

Grafico 1[1]: Procesamiento de Datos

Esquema de la conversión de los datos en información, mediante un procesamiento.

[1]Grafico 1 obtenido el 26 de octubre de 2010 en: http://es.wikipedia.org/wiki/Información

En el campo de la computación podemos decir que la información se la representa de varias maneras pero la principal información utilizada en este campo es la binaria por ser el lenguaje estándar para los ordenadores.

1.1.2 BASE DE DATOS

Una Base de Datos es la colección de datos los cuales son tratados y administrados como si fueran una sola unidad. La principal función de las Bases de Datos es almacenar y recuperar información de la misma.

En conclusión una Base de Datos no es más que la reunión de estructuras tanto lógicas como físicas que nos permiten depositar información de una manera ordenada y organizada, garantizándonos que dicha información está segura y disponible para quien la necesite y tenga los permisos para obtenerla.

1.1.3 SISTEMAS GESTORES DE BASES DE DATOS

Cuando hablamos de Sistemas Gestores de Bases de Datos (SGBD) nos referimos a él o los sistemas que controlan y administran las Bases de Datos, estos sistemas son los responsables de cómo se guarde y recupere la información de dichas bases de datos.

Son el intermediario entre el usuario y la Base de Datos, que mediante una interfaz le permite al usuario administrar adecuadamente la información ingresada.

1.2 ORIGEN DE LAS BASES DE DATOS[2]

Antes de aparecer los SGBD (década de los setenta), la información se trataba y se gestionaba utilizando los típicos sistemas de gestión de archivos que iban soportados sobre un sistema operativo. Éstos consistían en un conjunto de programas que definían y trabajaban sus propios datos. Los datos se almacenan en archivos y los programas manejan esos archivos para obtener la información. Si la estructura de los datos de los archivos cambia, todos los programas que los manejan se deben modificar; por ejemplo, un programa trabaja con un archivo de datos de alumnos, con una estructura o registro ya definido; si se incorporan elementos o campos a la estructura del archivo, los programas que utilizan ese archivo se tienen que modificar para tratar esos nuevos elementos. En estos sistemas de gestión de archivos, la definición de los datos se encuentra codificada dentro de los programas de aplicación en lugar de almacenarse de forma independiente, y además el control del acceso y la manipulación de los datos viene impuesto por los programas de aplicación.

Esto supone un gran inconveniente a la hora de administrar grandes volúmenes de información.

Surge así la idea de separar los datos contenidos en los archivos de los programas que los manipulan, es decir, que se pueda modificar la estructura de los datos de los archivos sin que por ello se tengan que modificar los

[2] Obtenido de: CEO - **Sistemas gestores de bases de datos**" del autor Mª. J. Ramos, A. Ramos, F. Montero, publicado por la editorial **McGraw-Hill** (Ramos y Montero s.f.)

programas con los que trabajan. Se trata de estructurar y organizar los datos de forma que se pueda acceder a ellos con independencia de los programas que los gestionan.

1.3 TIPOS DE BASES DE DATOS[3]

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, o la utilidad de la misma todo esto depende del fin que les demos a las Bases de Datos.

1.3.1 SEGÚN LA VARIABILIDAD DE LOS DATOS ALMACENADOS

1.3.1.1Bases de datos estáticas

Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

1.3.1.2 Bases de datos dinámicas

Éstas son bases de datos donde la información almacenada se modifica en el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de

[3]Obtenido el 4 de octubre de 2010 de: http://es.wikipedia.org/wiki/Base_de_datos

datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia o un videoclub.

1.3.2 SEGÚN EL CONTENIDO

1.3.2.1 Bases de datos bibliográficas

Solo contienen un representante de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque si no, estaríamos en presencia de una base de datos a texto completo Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio, entre otras.

1.3.2.2 Bases de datos de texto completo

Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

1.3.2.3 Directorios

Un ejemplo son las guías telefónicas en formato electrónico.

Bases de datos o "bibliotecas" de información química o biológica.

Son bases de datos que almacenan diferentes tipos de información proveniente de la química, las ciencias de la vida o médicas. Se pueden considerar en varios subtipos:

- Las que almacenan secuencias de nucleótidos o proteínas.
- Las bases de datos de rutas metabólicas.
- Bases de datos de estructura, comprende los registros de datos experimentales sobre estructuras 3D de biomoléculas.
- Bases de datos clínicas.

1.4 MODELOS DE BASES DE DATOS

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como *contenedor de datos* (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de *base de datos*; por lo general se refieren a algoritmos, y conceptos matemáticos.

1.4.1 Bases de datos jerárquicas

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un *nodo padre* de información puede tener varios *hijos*. El nodo que no tiene padres es llamado *raíz*, y a los nodos que no tienen hijos se los conoce como *hojas*.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

1.4.2 Base de datos de red

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de *nodo*: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

1.4.3 Bases de datos transaccionales

Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, estas bases son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial, es importante entender que su único fin es recolectar y recuperar los datos a la mayor velocidad posible, por lo tanto la redundancia y duplicación de información no es un problema como con las demás bases de datos, por lo general para poderlas aprovechar al máximo permiten algún tipo de conectividad a bases de datos relacionales.

1.4.4 Bases de datos relacionales

Éste es el modelo utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd[4], de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por *registros* (las filas de una tabla), que representarían las tuplas, y *campos* (las columnas de una tabla).

[4] Obtenido el 22 de octubre de 2010 en: http://www.dcc.uchile.cl/~rbaeza/inf/codd.html

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL[5], *Structured Query Language* o *Lenguaje Estructurado de Consultas*, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

Durante los años 80 la aparición de dBASE produjo una revolución en los lenguajes de programación y sistemas de administración de datos. Aunque nunca debe olvidarse que dBase no utilizaba SQL como lenguaje base para su gestión.

[5] SQL: Lenguaje declarativo de bases de datos relacionales que permite especificar diversos tipos de operaciones. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar información de manera sencilla. Obtenido de http://es.wikipedia.org/wiki/SQL el 28 de noviembre de 2010

1.4.5 Bases de datos multidimensionales

Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de Cubos OLAP[6]. Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos relacional podría serlo también en una base de datos multidimensional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar.

1.4.6 Bases de datos orientadas a objetos

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los *objetos* completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

[6] Obtenido el 15 de julio de 2010 de: http://www.dybox.cl/pdf/cubos.PDF

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

SQL: 2003, es el estándar de SQL92 ampliado, soporta los conceptos orientados a objetos y mantiene la compatibilidad con SQL92.

1.4.7 Bases de datos documentales

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes.

1.4.8 Bases de datos deductivas

Un sistema de base de datos deductiva, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. Las bases de datos deductivas son también llamadas bases de datos lógicas, a raíz de que se basa en lógica matemática.

1.4.9 Gestión de bases de datos distribuidas

La base de datos está almacenada en varias computadoras conectadas en red. Surgen debido a la existencia física de organismos descentralizados. Esto les da la capacidad de unir las bases de datos de cada localidad y acceder así por ejemplo a distintas universidades, sucursales de tiendas, etcétera.

CAPÍTULO 2: MEJORES PRÁCTICAS

2.1 Definiciones de Mejores Prácticas

Mejores prácticas o también llamado métodos correctos son enfoques o métodos que han demostrado su validez en la práctica.

Estas mejores prácticas son un respaldo sólido para las organizaciones que desean mejorar sus servicios, productos y/o procesos.

Según ITIL[7] "Por mejores prácticas se entiende como un conjunto coherente de acciones que han tenido éxito en un determinado contexto y que espera que en contextos similares, rindan similares resultados"

Según MBA Luis Ismodes[8] las mejores prácticas son: "Un Conjunto de procesos, actividades y recomendaciones, que una vez implantados, mejoran la eficacia y eficiencia de un negocio y que han sido probados en diferentes organizaciones, bajo diferentes condiciones y en múltiples oportunidades".

2.2 Mejores Prácticas Seguridad

2.2.1 Seguridad

La seguridad de la información no busca suprimir el riesgo por completo, la cual resultaría prácticamente imposible, sino gestionar la seguridad.

[7] ITIL (Information Technology Infrastructure Library) es un marco de trabajo de las mejores prácticas destinadas a facilitar la entrega de servicios de tecnologías de la información de alta calidad.
[8] http://www.tecnologiahechapalabra.com

Esta gestión del riesgo implica reducirlo, transferirlo o aceptarlo. A la hora de transferirlo, la opción más habitual es el aseguramiento de los activos más críticos con una compañía de seguros. Otra forma para transferir el riesgo, consiste en externalizar (outsourcing) su gestión a un proveedor de servicios de seguridad.

Los estudios sobre Seguridad y Delitos Informáticos de CSI/FBI 2005 han documentado que más del 70% de los ataques y la pérdida de datos de los sistemas de información han sido cometidos por integrantes de la organización, es decir, por aquellas personas por lo menos con autorización en algún nivel de acceso al sistema y sus datos. Las soluciones transparentes de seguridad son críticas en la actual economía global de negocios.

Históricamente la mayoría de las aplicaciones ha dependido de la seguridad en el nivel de aplicaciones para restringir el acceso a los datos sensibles. Los conceptos de seguridad como privilegio mínimo y necesidad de conocimiento fueron considerados menos importantes que la escalabilidad y la rápida implementación de nuevas aplicaciones. Internet aceleró el desarrollo de nuevas aplicaciones para todos los aspectos del procesamiento de negocios, dando como resultado una mejor accesibilidad, grandes ahorros de costo y aumentos de productividad. No obstante, las regulaciones mundiales ahora requieren controles más estrictos sobre información sensible y relacionada con la privacidad.

Es de vital importancia que se mantenga la seguridad en las áreas críticas de las Bases de Datos, entiéndase como áreas críticas a:

- Administración de Usuarios
- Control de Accesos
- Protección de datos
- Monitoreo de Datos

1) Al crear la base de datos existen gestores como Oracle en el cual se crean varios usuarios de administración automáticamente (*SYS*, *SYSTEM*, *SYSMAN*, DBSNMP), y otros muchos usuarios (SCOTT, HR, Anonymous, etc.) están creados pero tienen sus cuentas bloqueadas y no pueden conectarse. Las cuentas están bloqueadas por razones de seguridad, ya que un usuario mal intencionado, podría conectarse a la base de datos con las claves por defecto y consultar o modificar la información de la misma.

Por esto se recomienda que solamente se desbloqueen a los usuarios que vayan a ser utilizados, además sería recomendable eliminar las cuentas creadas por defecto en el momento de la creación de la Base de Datos, con el fin de garantizar la seguridad y mejorar el rendimiento de la Base de Datos al eliminar estas cuentas que no se van a utilizar.

Cuando hay muchos usuarios y objetos, la concesión de privilegios se hace pesada y tediosa, para simplificar esta tarea se han desarrollado los roles.

Un rol agrupa bajo un nombre una lista de privilegios, y puede ser asignado directamente a los usuarios. Este procedimiento se lo realiza para poder llevar a cabo un control mucho más riguroso de las capacidades y privilegios que tienen los usuarios que se conectan a la base de datos, ya que al no tener un diagrama de roles y privilegios se podrían cometer errores involuntarios en la asignación de permisos a información sensible para la empresa.

2) Es recomendable que se realice una adecuada segregación de funciones de manera que la persona que tiene los mayores privilegios de la base de datos, es decir el usuario SYS o SYSDBA en Oracle o SA en SQL Server no sea también aquella que posee la cuenta de mayor privilegio en el Sistema Operativo sobre el cual funciona la base de datos por ejemplo en Linux el usuario Root. Puesto que si dividimos estos roles conseguiremos administrar mejor la seguridad de la información que contiene la Base de Datos.

3) En la mayoría de Bases de Datos en la creación de usuarios siempre se pide el cambio de la contraseña la primera vez que el usuario se registra para acceder a la Base de Datos. No cambiar la contraseña inicial luego de la primera autenticación aumenta el riesgo de que usuarios no autorizados puedan autenticarse en la base de datos.

Además, aumenta el riesgo en la rendición de cuentas ya que no se podría tener certeza de las acciones de ese usuario, si el usuario y la persona que creó el identificador de usuario, conocen la contraseña.

4) Es importante controlar el acceso a la Base de Datos con el fin de resguardar la información que se maneje dentro del motor de Base de Datos, por lo cual es recomendable restringir el acceso mediante el uso de un usuario y contraseña u otros métodos más complejos y confiables de restricción de accesos (Firma Digital, Restricciones Bioelectrónicas, etc.) en el caso de usar restricción mediante usuario y contraseña una buena práctica para el control de accesos es la revisión de la tabla de usuarios en donde se tenga la contraseña de cada uno de dichos usuarios con el fin de verificar que todos y cada uno de los usuarios tengan asignada una contraseña, y así evitar que algún usuario pueda acceder sin un correcto registro. Esta actividad debe realizarla el personal autorizado en el departamento de TI.

5) Las contraseñas de acceso a la Base de Datos no deberían ser incluidas (Hard-Code[9]) en Scripts o líneas de comando los cuales no hayan sido revisados, monitoreados y asegurados de manera correcta con el fin de mantener la confidencialidad de la información.

6) El número de sesiones concurrentes debe ser limitado para todos los usuarios registrados en la Base de Datos. El administrador de la Base de Datos debería limitar el número de sesiones concurrentes para los usuarios y en la medida de lo posible manejar una sola sesión por usuario. Además es recomendable limitar la cantidad de tiempo que una sesión se encuentra

[9] Hard-Code: Mala práctica en el desarrollo e implementación de sistemas, en el cual se insertan datos en el código fuente.

inactiva antes de desconectarla de la Base de Datos, con el fin de reducir el riesgo que representa un ordenador con una sesión activa y que no se la esté utilizando ya sea porque el propietario de esa sesión olvidó cerrarla o porque se ausento de su lugar de trabajo. Lo cual puede provocar que dicha sesión sea usada por otra persona con fines perjudiciales o actividades no autorizadas.

7) La habilidad de permitir la modificación de las propiedades de los usuarios como por ejemplo:

La posibilidad de usar “GRANT OPTION” o “REVOKE” debe ser permitido solo a personal autorizado quienes sepan gestionar y administrar las propiedades de los demás usuarios.

8) La posibilidad de conectarse a la Base de Datos mediante una conexión remota debe ser permitida a ciertos host autorizados. Sobre todo esta restricción debe ser tomada en cuenta para los usuarios administradores (p.e. “Root”, “DBA”, “SYSDBA”) ya que son ellos quienes tienen mayores privilegios y pueden modificar cualquier parte crítica de la Base de Datos o Sistema Operativo, lo cual incrementa la posibilidad de actividades no autorizadas.

9) La información sensitiva como por ejemplo las contraseñas que son almacenadas en la Base de Datos deberían ser encriptadas. El no tener la información primordial encriptada aumenta el riesgo de accesos no

autorizados. La encripción ayuda a limitar el acceso a únicamente el personal que tienen los permisos de acceder a esta información confidencial.

10) Una de las mejores prácticas que se han establecido y ha sido probada en varias ocasiones y en varios lugares del mundo, pero así mismo una de las que menos se ocupa ya que su aplicación afecta directamente al rendimiento y procesamiento de la Base de Datos. Es la activación de los registros de auditoría con lo cual se logra un monitoreo y control de las actividades que se realizan en la Base de Datos, con la utilización de esta recomendación se puede conocer que persona realizó qué actividad en específico, cuando y desde que equipo de computación.

Uno de los principales justificativos para no utilizar esta funcionalidad que nos ofrecen la mayoría de Motores de Bases de Datos ya sean estos de arquitectura comercial como de arquitectura libre, es la disminución del rendimiento de la Base de Datos al momento de activar este control, pero hoy en día la mayoría de los motores de Bases de datos permiten focalizar el registro de auditoría a ciertos objetos o tablas de la base de datos, es decir que podemos seleccionar que actividades deseamos que sean registrados en el log de auditoría.

11) El administrador de seguridad (no el DBA) debe supervisar la configuración de auditoría de base de datos. Sólo el administrador de seguridad debe tener acceso a los registros de auditoría. Si los usuarios privilegiados, como los administradores de bases que se autentican a la

base de datos directamente, pueden acceder a los registros de auditoría y cambiar la configuración establecida sin la debida supervisión, por los cual este o estos usuarios pueden pasar por alto la auditoría y realizar cambios no autorizados sin que sea posible hacerlos responsables.

12) Se debería asegurar que los controles de Respaldo y Restauración de la Base de Datos garantizan la disponibilidad de los datos los mismos que se pueden recuperar por completo, con precisión y de manera oportuna.

La falta de copias de seguridad de archivos de base de datos, podría resultar en una incapacidad para recuperarse plenamente en situaciones de riesgo.

13) Para poder asegurar la existencia de los datos en el momento en que se los necesite se debe garantizar que solo los usuarios permitidos puedan alterar, visualizar o eliminar la información almacenada en la base.

14) Cada uno de los identificadores de usuario deben ser únicos e identificar un único usuario, respetando la norma de denominación corporativa para este punto.

Existe un riesgo mayor si los IDs de usuario podrían estar siendo compartidos. La función o dueño de un ID de usuario será difícil de identificar si no se sigue la convención de nomenclatura estándar. Por ejemplo:

Si el nombre del empleado es Roberto Burbano su ID correspondiente alineada a la norma de la corporación la cual dice que el usuario de cada empleado esta dado por la primera letra de su nombre seguido de su

apellido más un numero secuencial, entonces el identificador sería RBURBANO001.

15) Un usuario con acceso de línea de comandos podría eludir controles y seguridades de las aplicaciones de bases de datos, él / ella podría ejecutar o modificar archivos binarios de la configuración de la base de datos.

Los usuarios con acceso a scripts SQL pueden usar las secuencias de comandos para realizar actividades no autorizadas en la base de datos ya que estos suelen contener secuencias de comandos de base de datos de identificadores de usuario y contraseñas.

Por tanto, la integridad, disponibilidad y confidencialidad de la base de datos podría verse comprometida.

2.3 Mejores Prácticas de Integridad

2.3.1 Integridad

Según Gonzalo Álvarez Marañón y Pedro Pérez García autores del libro "Seguridad Informática para empresas y particulares" dicen "La Integridad consiste en garantizar que los datos, objetos y recursos no han sido alterados, permanecen completos y son fiables".

Entonces podemos decir que la integridad de una base de datos significa que, la base de datos o los programas que generaron su contenido,

incorporen métodos que aseguren que el contenido de los datos del sistema no se rompan así como las reglas del negocio. Por ejemplo, un distribuidor puede tener una regla la cual permita que sólo los clientes individuales puedan solicitar órdenes; a su vez cada orden identifique a uno y sólo un proveedor. El servidor Oracle y otros SGDB relacionales hacen cumplir este tipo de reglas del negocio con limitantes, las cuales pueden ser configuradas implícitamente a través de consultas. Para continuar con este ejemplo, en el proceso de inserción de una nueva orden en la base de datos, está tendría que cerciorarse de que el cliente identificado existe en su tabla para que la orden pueda darse.

El objetivo de la Integridad consiste en garantizar que los datos, objetos y recursos no han sido alterados, permanecen completos y son fiables. La modificación no autorizada de los datos no puede ocurrir durante su almacenamiento, transporte o procesamiento. Por tanto, se vuelve necesario implantar mecanismos de control de integridad durante todos los estados de la información.

La principal y más básica de las practicas que nos ayudan a mantener y garantizar la integridad de la información almacenada en la Base de datos que administremos en nuestra organización, es la aplicación de Vistas con el fin de obtener la información ya sea de una o varias tablas.

Por lo cual necesitamos definir una Vista[10] en una Base de Datos.

[10] Definición de vista obtenida de: http://msdn.microsoft.com/es-es/library/ms190174.aspx El 28 de noviembre de 2010

“Una vista es una tabla virtual cuyo contenido está definido por una consulta. Al igual que una tabla real, una vista consta de un conjunto de columnas y filas de datos con un nombre. Sin embargo, a menos que esté indexada, una vista no existe como conjunto de valores de datos almacenados en una base de datos. Las filas y las columnas de datos proceden de tablas a las que se hace referencia en la consulta que define la vista y se producen de forma dinámica cuando se hace referencia a la vista”.

Una vista actúa como filtro de las tablas subyacentes a las que se hace referencia en ella. La consulta que define la vista puede provenir de una o de varias tablas, o bien de otras vistas de la base de datos actual u otras bases de datos. Así mismo, es posible utilizar las consultas distribuidas para definir vistas que utilicen datos de orígenes heterogéneos. Esto puede resultar de utilidad, por ejemplo, si desea combinar datos de estructura similar que proceden de distintos servidores, cada uno de los cuales almacena los datos para una región distinta de la organización.

No existe ninguna restricción a la hora de consultar vistas y muy pocas restricciones a la hora de modificar los datos de éstas. Ver gráfico 2.

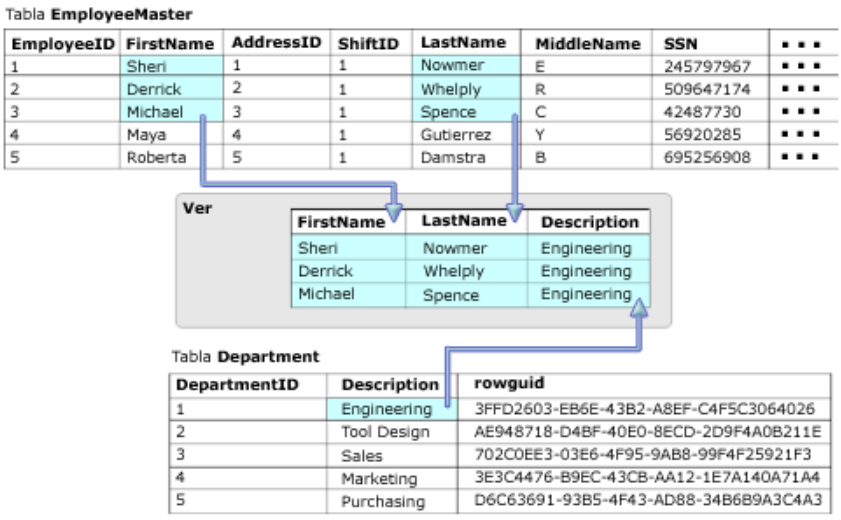
En otras palabras, una vista es una ventana a través de la cual se puede consultar o cambiar información de la tabla a la que está asociada. Esto, claro está, en relación con los privilegios que posea el usuario de la base de datos. Si el usuario solamente tiene privilegios de lectura en una entidad, en la vista tampoco podrá agregar o modificar información; si el usuario no tiene

acceso a determinadas tablas, tampoco podrá crear una vista con información proveniente de las mismas.

Las vistas tienen la misma estructura que una tabla: filas y columnas. La única diferencia es que sólo se almacena de ellas la definición, no los datos. Los datos que se recuperan mediante una consulta en una vista se presentarán igual que los de una tabla. De hecho, si no se sabe que se está trabajando con una vista, nada hace suponer que es así. Al igual que sucede con una tabla, se pueden insertar, actualizar, borrar y seleccionar datos.

Esto significa que una vista no contiene datos duplicados de una tabla de la base de datos. No tiene absolutamente ningún dato de la tabla real, pues como ya se mencionaba, no es una tabla real. Es decir, se percibe como una tabla virtual. Ver Gráfico 2.

Gráfico 2[11] Diagrama de una Vista



La vista es creada a partir de 2 tablas EmployeeMaster y Department con la visualización de los campos FirstName, LastName y Description. Con ciertas clausulas para no visualizar todos los datos.

[11] Gráfico 2 nos ilustra la obtención de una vista a partir de 2 tablas. Obtenido de: http://msdn.microsoft.com/es-es/library/ms190174.aspx El 21 de septiembre de 2010

Consecuentemente las vistas nos ayudan en varios aspectos entre los cuales encontramos:

- Las vistas pueden proporcionar un nivel adicional de seguridad.
- Las vistas permiten ocultar la complejidad de los datos. Una base de datos se compone de muchas tablas. La información de dos o más tablas puede recuperarse utilizando una o varias combinaciones de tablas, y estas combinaciones pueden resultar muy confusas. Creando una vista se hace todo visualmente más simple. Y así generamos un mayor control de integridad.
- Las vistas ayudan a mantener nombres razonables para las consultas.

Otro mecanismo que se aplica en las Bases de datos para garantizar la integridad de la información es la administración de privilegios mediante roles para garantizar que solo los usuarios autorizados puedan modificar, eliminar e incluso ingresar la información que se encuentra en la base de datos. Es necesario definir que usuarios tienen permisos para las diferentes actividades, lo que implica algún tipo de manipulación de la información almacenada.

Por lo cual es recomendable activar los registros de auditoría para las diferentes sentencias que permiten la modificación de información en la base de datos. Por ejemplo en una institución financiera en donde la manipulación de la información puede causar grandes pérdidas tanto económicas como

institucionales, para la compañía es necesario activar la auditoria de las sentencias tales como Insert, Update y Delete. De esta manera poder llevar un seguimiento y control sobre las operaciones que se realicen y/o modifiquen la integridad de los datos.

Una definición que nos puede ayudar a mantener la integridad de los datos es la utilización del concepto de integridad referencial que es: "Un sistema de reglas que utilizan la mayoría de las bases de datos relacionales para asegurarse que los registros de tablas relacionadas son válidos y que no se borren o cambien datos relacionados de forma accidental produciendo errores de integridad."[12]

En el siguiente ejemplo se utilizan 2 tablas las cuales se encuentran relacionadas. Cuando se define una columna como clave foránea, las filas de la tabla pueden contener en esa columna o bien el valor nulo (ningún valor), o bien un valor que existe en la otra tabla, un error sería asignar a un habitante una población que no está en la tabla de poblaciones. Eso es lo que se denomina integridad referencial y consiste en que los datos que referencian otros (claves foráneas) deben ser correctos. La integridad referencial hace que el sistema gestor de la base de datos se asegure de que no existan en las claves foráneas valores que no estén en la tabla principal.

[12] http://www.aulaclic.es/sql/b_8_1_1.htm

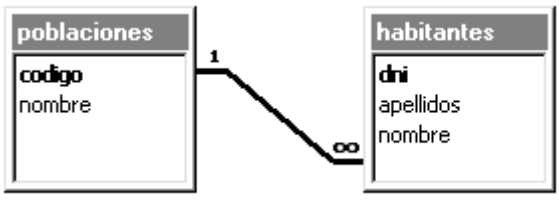
La integridad referencial se activa cuando creamos una clave foránea y a partir de ese momento se comprueba cada vez que se modifiquen datos que puedan alterarla.

La Integridad referencial vigila que se cumplan las siguientes reglas:

- No se podrá introducir un valor en la tabla relacionada si antes no ha sido introducida en la tabla principal.
- No se puede eliminar un registro de una tabla principal si existen registros coincidentes en la tabla relacionada.
- No se puede cambiar un valor de la clave primaria en la tabla principal si el registro tiene registros relacionados.

Como ejemplo nos plantearemos la siguiente unión entre dos tablas relacionadas de “uno a muchos”

Gráfico 3[13] Relación de Tablas



DNI: Documento Nacional de Identidad

[13] Gráfico y explicación obtenidos de http://www.aulaclic.es/sql/b_8_1_1.htm El 20 de agosto de 2010

Ejemplos de Errores en Los Datos

Para el presente ejemplo se utilizará la siguiente Información

Poblaciones	
Código	Nombre
1	Pichincha
2	Esmeraldas

Habitantes		
DNI	Apellidos	Nombres
1719711702	Burbano	Marco
1789367492	Guerrero	Grace

- Cuando insertamos una nueva fila en la tabla secundaria y el valor de la clave foránea no existe en la tabla principal. Insertamos un nuevo habitante y en la columna población escribimos un código de población que no está en la tabla de poblaciones (una población que no existe).

- Cuando modificamos el valor de la clave principal de un registro que tiene 'hijos', modificamos el código de Pichincha, sustituimos el valor que tenía (1) por un nuevo valor (10), si Pichincha tenía habitantes asignados, qué pasa con esos habitantes, no pueden seguir teniendo el código de población 1 porque la población 1 ya no existe, en este caso hay dos alternativas, no dejar cambiar el código de Pichincha o bien cambiar el código de población de todos los habitantes de Pichincha y asignarles el código 10.

- Cuando modificamos el valor de la clave foránea, el nuevo valor debe existir en la tabla principal. Por ejemplo cambiamos la población de un habitante, tenía asignada la población 1 (porque estaba empadronado en Pichincha) y ahora se le asigna la

población 2 porque cambia de lugar de residencia. La población 2 debe existir en la tabla de poblaciones.

- Cuando queremos borrar una fila de la tabla principal y ese registro tiene 'hijos', por ejemplo queremos borrar la población 1 (Pichincha) si existen habitantes asignados a la población 1, estos no se pueden quedar con el valor 1 en la columna población porque tendrían asignada una población que no existe. En este caso tenemos dos alternativas, no dejar borrar la población 1 de la tabla de poblaciones, o bien borrarla y poner valor nulo el campo población de todos sus 'hijos'.

Asociada a la integridad referencial están los conceptos de actualizar los registros en cascada y eliminar registros en cascada. Pero estos temas no son parte del desarrollo de este proyecto.

2.4 Mejores Prácticas de Disponibilidad

2.4.1 Disponibilidad[14]

Según Gonzalo Álvarez Marañón y Pedro Pérez García autores del libro "Seguridad Informática para empresas y particulares" dicen "Disponibilidad consiste en garantizar que los datos permanecen accesibles sin interrupciones cuando y donde se los necesita".

[14] Obtenido el 28 de noviembre de 2010 de http://es.wikipedia.org/wiki/Administrador_de_base_de_datos

Entonces podemos decir que disponibilidad significa que los usuarios autorizados tengan acceso a los datos cuando los necesiten para atender a las necesidades del negocio. De manera incremental los negocios han ido requiriendo que su información esté disponible todo el tiempo (“24/7”, veinticuatro horas del día, los siete días a la semana). La industria de TI ha respondido a estas necesidades con redundancia de red y hardware para incrementar las capacidades administrativas en línea.

El objetivo de la disponibilidad consiste en garantizar que los datos permanecen accesibles sin interrupciones cuando y donde sea que se los necesita. La disponibilidad exige que se implanten una serie de controles para asegurar un nivel razonable de rendimiento, una gestión rápida y eficiente de las interrupciones, proporcionar replicación continua, mantener copias de seguridad actualizadas y evitar la pérdida o destrucción de datos.

Con frecuencia, las amenazas contra la disponibilidad poseen un origen más fortuito que deliberado: fallos en el hardware o en la alimentación eléctrica, errores en el software, condiciones ambientales extremas, como calor, frio o humedad excesivos, servicios infradimensionados que no son capaces de atender a todas las peticiones, usuarios y administradores negligentes.

No cabe duda que al hablar de disponibilidad una de las principales soluciones que se presentan es la clusterización lo cual se define como:

“Los conjuntos o conglomerados de computadoras construidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora.

Hoy en día desempeñan un papel importante en la solución de problemas de las ciencias, las ingenierías y del comercio moderno.

La tecnología de clúster ha evolucionado en apoyo de actividades que van desde aplicaciones de supercómputo y software de misiones críticas, servidores web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos.”[15]

Otro punto muy importante para el manejo de la disponibilidad en un gestor de base de datos, es la utilización de redundancia. Por lo que es necesario definir el concepto de redundancia en bases de datos.

“Se llama redundancia al hecho de que los mismos datos estén almacenados más de una vez en la base de datos. Las redundancias además de suponer un consumo de recursos de almacenamiento pueden llevar a situaciones en las que un dato se actualice en una de sus ubicaciones y en otra no y se pierda la integridad de la BD, por tanto deben evitarse.”[16]

A primera vista la redundancia no parece más que un problema en las bases de datos ya que esta disminuiría el performance de la base de datos así

[15] Ver más http://es.wikipedia.org/wiki/Cluster_(informática) obtenido el 20 de agosto de 2010
[16] Ver más http://www.csae.map.es/csi/silice/Sgbdat6.html obtenido el 25 de agosto de 2010

como utilización de recursos tanto físicos como lógicos además de presentarnos problemas de integridad de la información almacenada.

Es por esto que se ha definido una variante de redundancia. La cual se denomina redundancia controlada, lo cual es: La utilización o introducción de redundancia a un gestor de base de datos de manera intencional y con pleno conocimiento de esto, tomando en cuenta la utilización de recursos y optimizando la duplicación de información lo cual permite incrementar la eficiencia de disponibilidad.

¿Cómo se aumenta la disponibilidad al insertar redundancia controlada? Primeramente se necesitaría instalar varios servidores en vez de un solo servidor de base de datos. Estos varios servidores deben poseer la capacidad de trabajar en paralelo y principalmente de asumir el trabajo de alguno de ellos que sufra una caída inesperada (disponibilidad) a lo cual conocemos como un clúster de servidores.

De esta manera podemos decir que el uso de un clúster el cual ha sido configurado incluyendo redundancia controlada nos ayuda a mejorar la disponibilidad de nuestro gestor de base de datos. Cabe mencionar que la utilización de clúster y redundancia no es aplicable a todos los casos que se presenten en las diferentes compañías y/o empresas que deseen aumentar la disponibilidad de su respectiva información Pero podemos decir que cuando la cantidad de información es considerable así como también cuando la información es de vital uso para las diferentes actividades de la

empresa como por ejemplo la información de las cuentas que maneja un banco es de vital importancia para el funcionamiento del mismo.

Otra solución que se presenta para el incremento de disponibilidad es la vinculación del servicio de un banco de información ya que como se ha definido previamente la información que posee una empresa se ha constituido en un bien o activo fijo que debe ser resguardado de manera adecuada para así garantizar la obtención de esta información en el momento que se la requiera.

Pese a que la investigación sobre Bancos de Información no es parte del presente trabajo es necesario conocer que es un Banco de información, como trabaja y como brinda sus servicios.

Aquí nace un nuevo concepto el cual es el Banco de Información lo que no difiere mucho de un banco o institución financiera. Ya que en el banco de información se deposita información y se la puede recuperar mediante consultas a la información almacenada en el banco. Se trataría de una tercera empresa la cual se encargue completamente del resguardo de la información de la organización.

Con la implementación de bancos de información la empresa asegura la información de manera que esté completamente disponible en el momento que se la necesita.

Cabe mencionar que con la ventaja brindada por la contratación de un banco de información la empresa delega la responsabilidad a una empresa especializada en el almacenamiento de grandes cantidades de información.

Lo cual representa una preocupación menos para la empresa contratante del servicio antes mencionado.

2.5 Obtención de las Mejores Prácticas

Las mejores prácticas tanto de seguridad, integridad y disponibilidad que se han descrito anteriormente y que serán aplicadas más adelante han sido obtenidas de diferentes fuentes de donde se ha consultado y consolidado las principales y más importantes actividades para la configuración de Bases de Datos, con la finalidad que mantengan a lo largo de su ciclo de vida los objetivos de seguridad, integridad y disponibilidad que este trabajo pretende dar a conocer.

Entre las principales fuentes consultadas tenemos los Framework de COBIT e ITIL los cuales se enfocan en la definición de mejores prácticas para entregar valor tanto en la administración de la Tecnología como en la administración del negocio.

Otra fuente de conocimiento que ha alimentado el desarrollo de este proyecto ha sido la experiencia laboral del autor, puesto que en el trabajo diario realiza diferentes actividades de evaluación y auditoria de sistemas informáticos y bases de datos tanto comerciales como de software libre.

Finalmente entre las principales fuentes que ayudaron en gran medida la finalización del proyecto han sido los manuales, papers, y materiales de certificación tanto físicos como digitales expuestos en su mayoría por los

creadores y desarrolladores de Oracle y MYSQL de donde se ha obtenido la solución técnica de la implementación de cada uno de los motores de Bases de Datos.

2.5.1 Resumen Mejores Prácticas

Tabla 1: Resumen de las Mejores Prácticas de Seguridad, Integridad y Disponibilidad en Base de Datos, implementadas en Oracle y MYSQL.

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL
1.	Se debe mantener bloqueados a los usuarios creados automáticamente por la BDD y en el mejor de los casos se los debe Borrar con la finalidad de mantener la seguridad.	Seguridad	En Oracle se mantienen bloqueados hasta que un usuario con permiso los libere y permita su funcionamiento	MySql crea automáticamente el usuario administrador(ROOT), el cual es necesario para el correcto funcionamiento de la BDD

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL
2.	Se debe realizar segregación de funciones en el SO y La BDD con la finalidad de no colocar en una sola persona la completa seguridad del ambiente en el que se encuentra la BDD	Seguridad	Se lo realiza evitando una autenticación en la BDD mediante la autenticación del SO y para una mayor seguridad las claves de mayor poder en La BDD y SO deben pertenecer a usuarios diferentes	Se lo realiza evitando una autenticación en la BDD mediante la autenticación del SO y para una mayor seguridad las calves de mayor poder en La BDD y SO deben pertenecer a usuarios diferentes
3.	Mediante aplicación o si la BDD lo permite pedir el cambio de contraseña obligatorio después de la creación de un usuario	Seguridad	Oracle lo ha implementado automáticamente incluso se puede definir una periodicidad para solicitar cambios de contraseñas	Se lo puede controlar mediante una aplicación que funcione con la BDD, así como también automatizar la petición de cambio de

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL
				contraseña.
4.	Restringir el acceso de usuarios autorizados a la BDD	Seguridad	Se debe definir un método de acceso a la BDD el cual puede ser mediante el uso de un usuario y contraseña autorizados o cualquier otro método eficaz	Se debe definir un método de acceso a la BDD el cual puede ser mediante el uso de un usuario y contraseña autorizados o cualquier otro método eficaz
5.	Para la ejecución de scripts o líneas de comando estas deben tener previa autorización más aún si los escripts tienen incluidas	Seguridad	Definir que scripts y con la autorización de quien dentro del departamento de TI son los autorizados para ser ejecutados en la BDD	Definir que scripts y con la autorización de quien dentro del departamento de TI son los autorizados para ser ejecutados en la BDD

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL
	contraseñas de usuarios con poder en la BDD			
6.	Limitar el número de sesiones concurrentes por cada usuario de preferencia permitir una sola sesión por cada usuario	Seguridad	Oracle permite esta restricción	MySql permite esta restricción
7.	La posibilidad de usar “GRANT OPTION” o “REVOKE” debe ser permitido solo a personal autorizado	Seguridad	Oracle permite esta restricción	MySql permite esta restricción

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL
8.	Denegar la posibilidad de conectarse remotamente a la BDD más aún si los usuarios tienen permisos para modificar la información sensitiva	Seguridad	Oracle permite esta restricción	MySql permite esta restricción
9.	La información sensitiva almacenada en la Base de Datos debería ser encriptada. El no tener la información primordial encriptada aumenta el riesgo de accesos no autorizados.	Seguridad	Oracle permite esta restricción	MySql permite esta restricción

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL
10.	Activar los registros de auditoría con la finalidad de llevar un control sobre las actividades realizadas en la BDD	Seguridad	Oracle permite esta restricción	Se lo debería implementar mediante un Trigger o un aplicativo
11.	El administrador de seguridad (no el DBA) debe supervisar la configuración de auditoría de base de datos. Sólo el administrador de seguridad debe tener acceso a los registros de auditoría.	Seguridad	Se lo implementa mediante roles y permisos	Se lo implementa mediante roles y permisos

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL
12.	Asegurar que los controles de Respaldo y Restauración de la Base de Datos garantiza la disponibilidad de los datos los cuales se pueden recuperar por completo	Seguridad/ Disponibilidad	Oracle permite ejecutar respaldos la revisión va por cuenta del departamento de IT	MySql permite ejecutar respaldos la revisión va por cuenta del departamento de IT
13.	Realizar una correcta asignación de privilegios a cada uno de los usuarios	Seguridad/ Integridad	Implementar procedimientos rigurosos para la asignación de perfiles y roles en la BDD	Implementar procedimientos rigurosos para la asignación de perfiles y roles en la BDD
14.	Todos los identificadores de usuario debe ser único e identificar un único usuario y debe respetar la norma de denominación corporativa.	Seguridad	Definir políticas para el acceso a la BDD	Definir políticas para el acceso a la BDD
15.	Restringir el uso de líneas de	Seguridad	Restricción del SO	Restricción del SO

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL
	comando a usuarios autorizados			
16.	Aplicación de vistas con la finalidad de restringir la información a usuarios finales	Integridad	Oracle permite la realización de Vistas	MySql permite la realización de Vistas
17.	Aplicar integridad referencial	Integridad	Implementado en Oracle	Implementado en MySql con Tablas INNODB
18.	Utilización de clúster para mejorar la disponibilidad	Disponibilidad	Oracle permite la realización de clúster mediante una configuración en el Motor de	MySQL no tiene esta función integrada, pero existe la posibilidad de

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL
			BDD	configurar clúster de manera manual.
19.	Utilización del servicio de un Banco de Datos	Disponibilidad	No depende de la BDD local	No depende de la BDD local
20.	Utilización de redundancia	Disponibilidad	Oracle permite generar redundancia	Se lo puede realizar de manera manual o mediante una aplicación.

CAPÍTULO 3: IMPLANTACIÓN DE LAS BASES DE DATOS UTILIZANDO LAS MEJORES PRÁCTICAS

3.1 Antecedentes de Implementación

Para realizar pruebas sobre las técnicas mencionadas anteriormente se realizará la implementación de un modelo de Base de Datos sobre dos arquitecturas diferentes las cuales son:

- Arquitectura Comercial: Para esta arquitectura se ha escogido la BDD ORACLE como la principal representante de esta arquitectura.
- Arquitectura Libre Para la arquitectura libre se ha escogido la BDD MYSQL puesto que ha tenido una gran acogida en el mercado.

Cabe mencionar que las bases de datos serán instaladas sobre un Sistema Operativo de la Plataforma Microsoft en la Versión Windows Seven Ultimate de 32 bits.

Para probar las técnicas descritas en el capítulo anterior de este se ha descrito un modelo “Entidad – Relación” para ser implementado en las dos bases de datos seleccionadas.

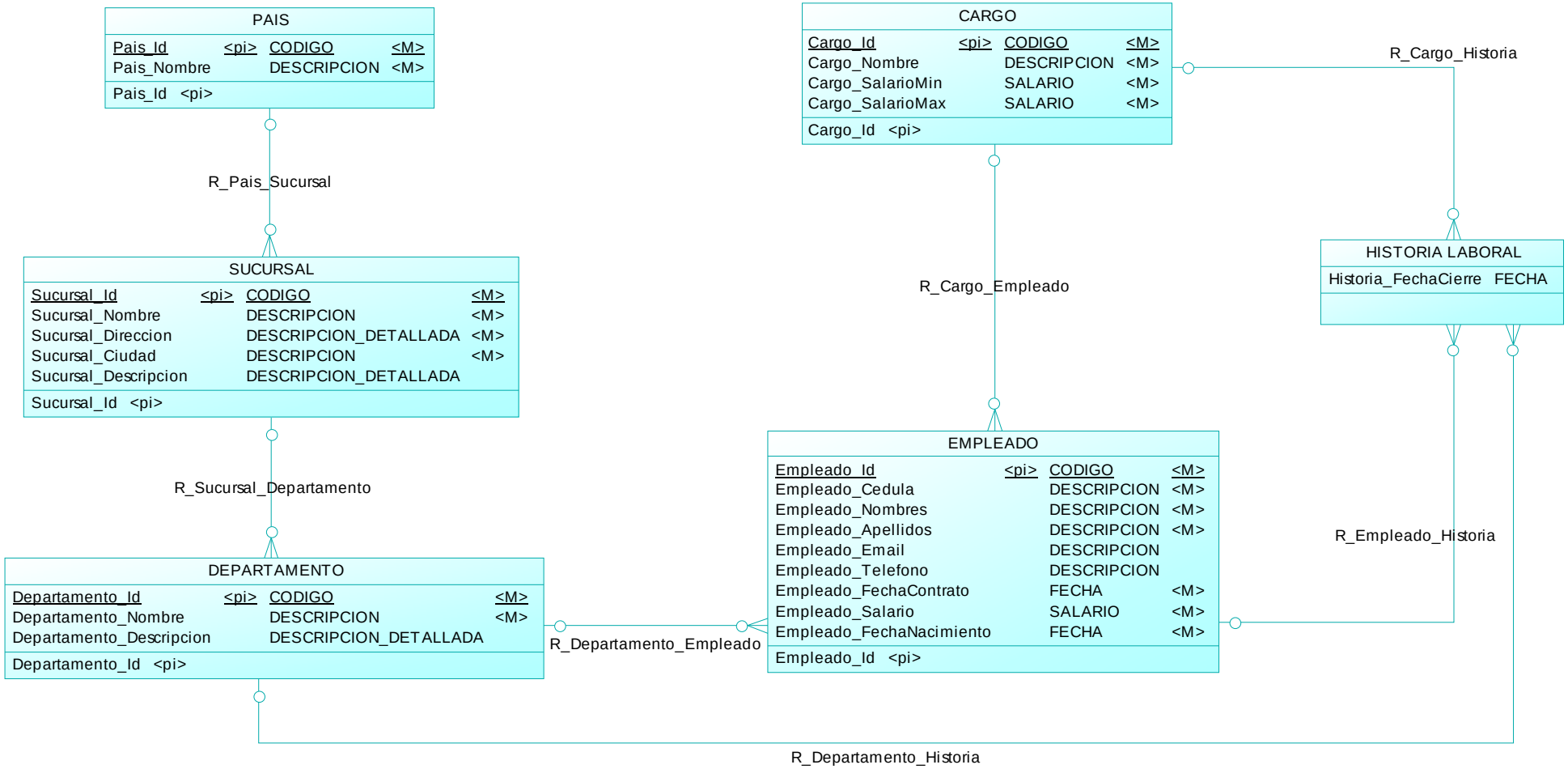
El modelo trata sobre la estructura administrativa del Talento Humano (Recursos Humanos) de una empresa que tiene sede en varios países. Que a su vez puede o no tener varias sucursales por cada país. Cabe mencionar que cada sucursal tiene varios departamentos en donde se desarrollan varios cargos. Dichos cargos son desarrollados por uno varios empleados.

Finalmente se tiene un registro de la historia laboral de cada uno de los empleados en donde se encuentra información relacionada a fecha de contratación los cargos desempeñados así como también a que departamento pertenece cada cargo.

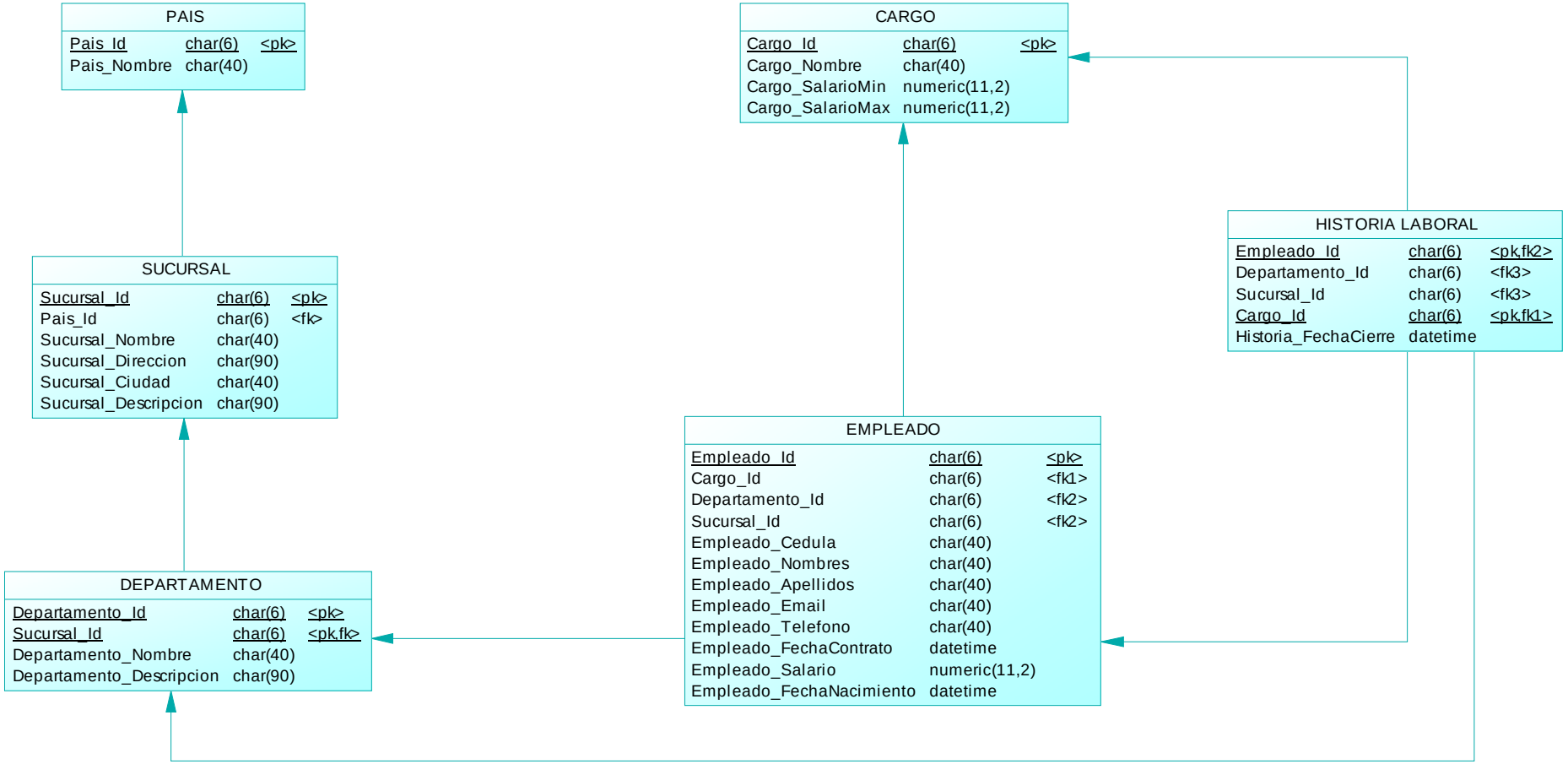
3.1.1 Modelado

Para el diseño del modelo “Entidad – Relación” se utilizó la herramienta gráfica “PowerDesigner 15.1” con la cual se desarrolló un Modelo Conceptual y un Modelo Físico que posteriormente se lo impactó a cada una de las bases de datos.

3.1.1.1 Modelo Conceptual:



3.1.1.2 Modelo Físico:



3.2 Implementación en una Base de Datos de Arquitectura Libre (MYSQL)

1. **Se debe mantener bloqueados a los usuarios creados automáticamente por la BDD y en el mejor de los casos se los debe Borrar con la finalidad de mantener la seguridad.**

Implementación

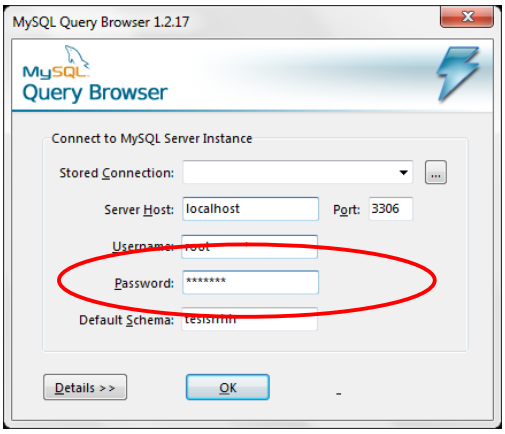
MySql crea automáticamente el usuario administrador (ROOT), el cual es necesario para el correcto funcionamiento de la BDD, no existen más usuarios creados por defecto. En este punto MYSQL no presente debilidad alguna.

2. **Se debe realizar segregación de funciones en el SO y La BDD con la finalidad de no colocar en una sola persona la completa seguridad del ambiente en el que se encuentra la BDD**

Implementación

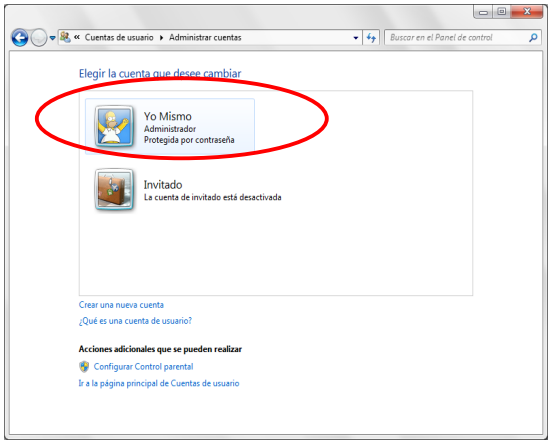
Se lo realiza evitando una autenticación en la BDD mediante la autenticación del SO y para una mayor seguridad las claves de mayor poder en La BDD y SO deben pertenecer a usuarios diferentes

Autenticación en la Base de Datos



Realizado por Marco Burbano Fecha: 2010/05/15

Autenticación en el SO mediante una contraseña.



Realizado por Marco Burbano Fecha: 2010/05/15

3. Mediante aplicación o si la BDD lo permite pedir el cambio de contraseña obligatorio después de la creación de un usuario

Implementación

MYSQL no otorga la posibilidad de administrar el cambio de contraseña después de la creación de un usuario, pero esto se lo puede implementar a través de una aplicación que funcione conjuntamente con este motor de base de datos.

4. Restringir el acceso de usuarios autorizados a la BDD

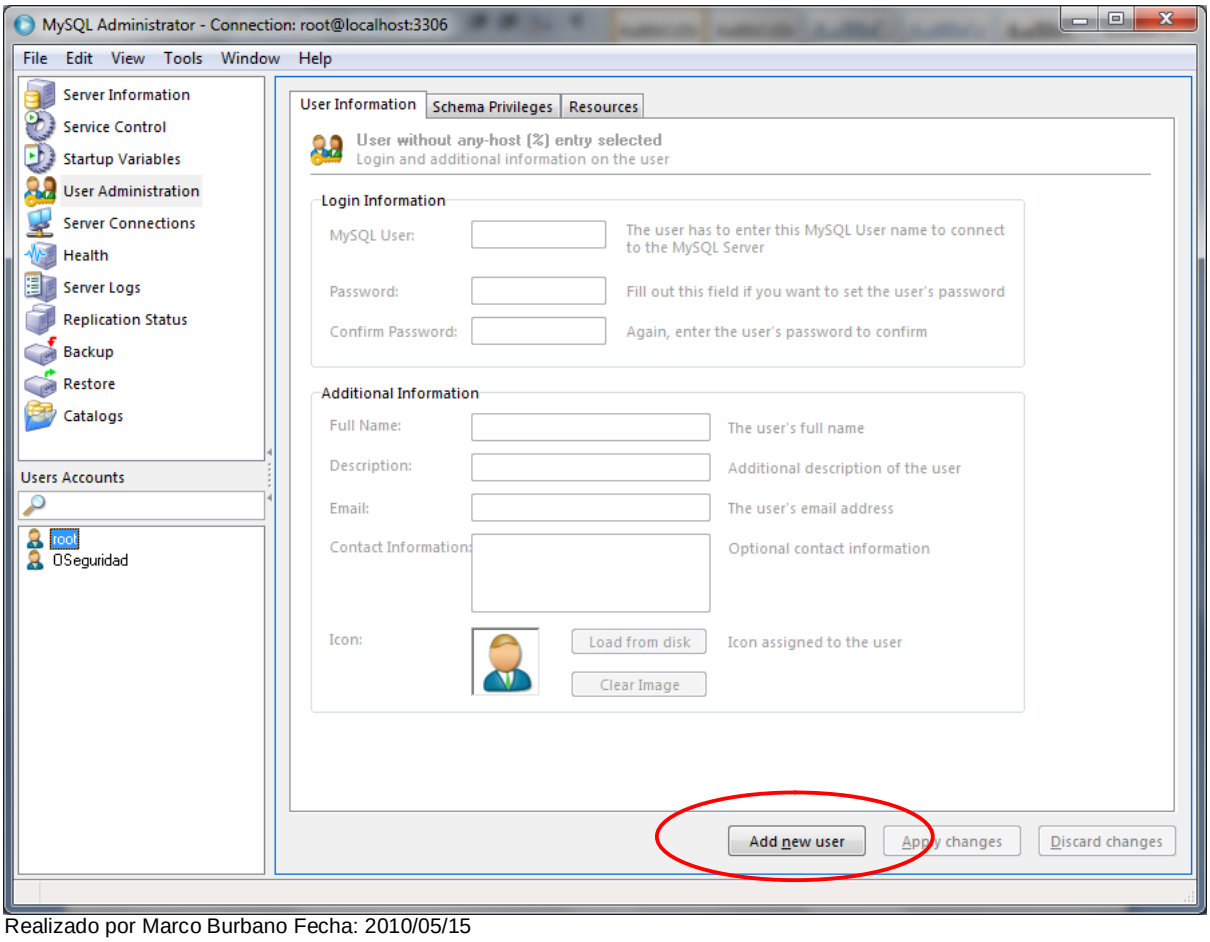
Implementación

Se debe definir un método de acceso a la BDD el cual puede ser mediante el uso de un usuario y contraseña autorizados o cualquier otro método eficaz.

En MYSQL lo más común es el acceso autorizado mediante la asignación de usuario y contraseña

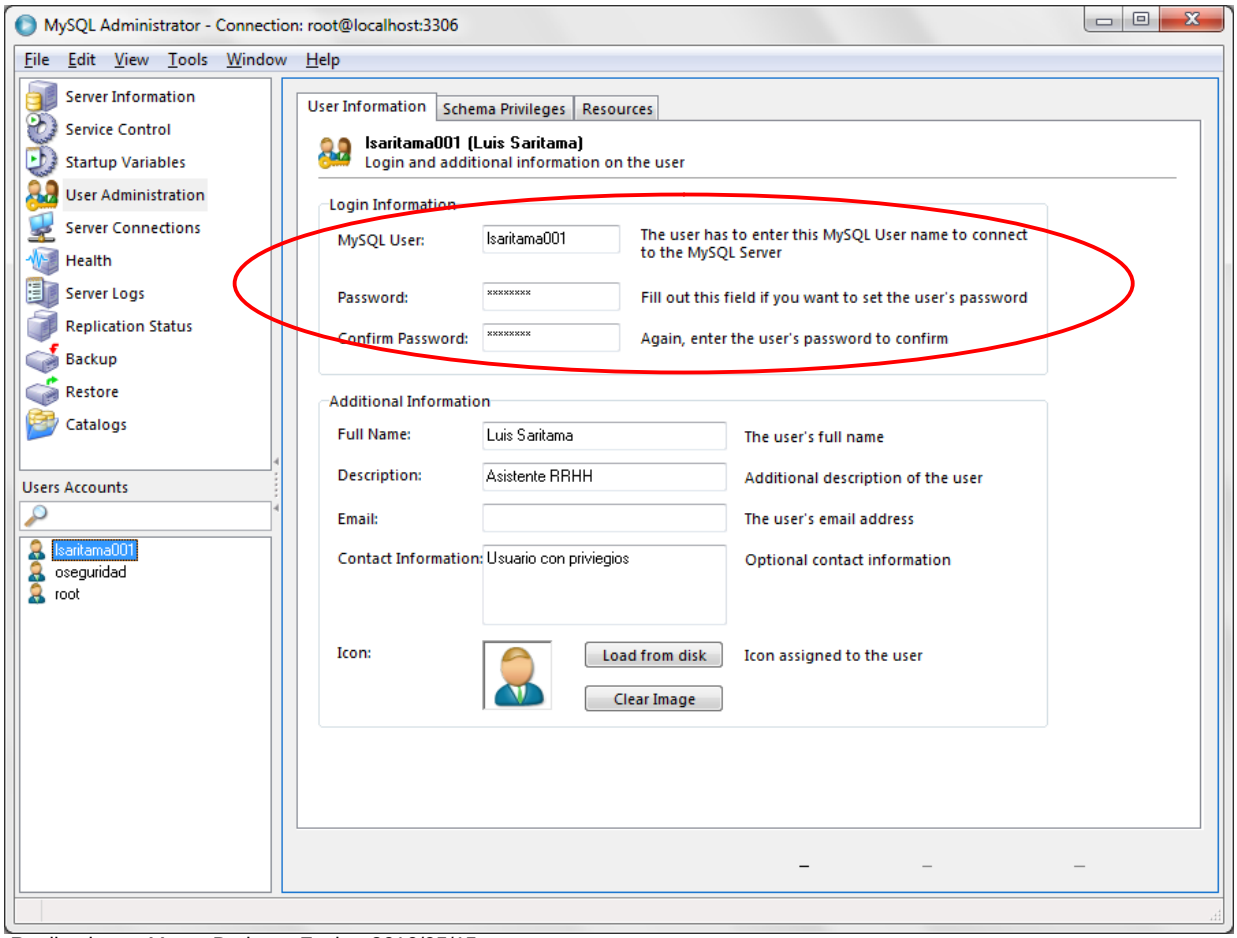
Para esta práctica podemos utilizar la interfaz gráfica del MYSQL Administrator.

Inicialmente debemos realizar la creación de un nuevo usuario.



Realizado por Marco Burbano Fecha: 2010/05/15

En donde posteriormente definiremos el usuario y contraseña que le asignaremos para poder conectarse a la base de datos.



Realizado por Marco Burbano Fecha: 2010/05/15

5. **Para la ejecución de scripts o líneas de comando estas deben tener previa autorización más aún si estos tienen incluidas contraseñas de usuarios con poder en la BDD**

Implementación

Se debe definir políticas de aprobaciones para ejecutar scripts y con qué usuario y contraseña deben ser ejecutados además se debe definir quién es el responsable de la ejecución y revisión de estas rutinas. Así como también se

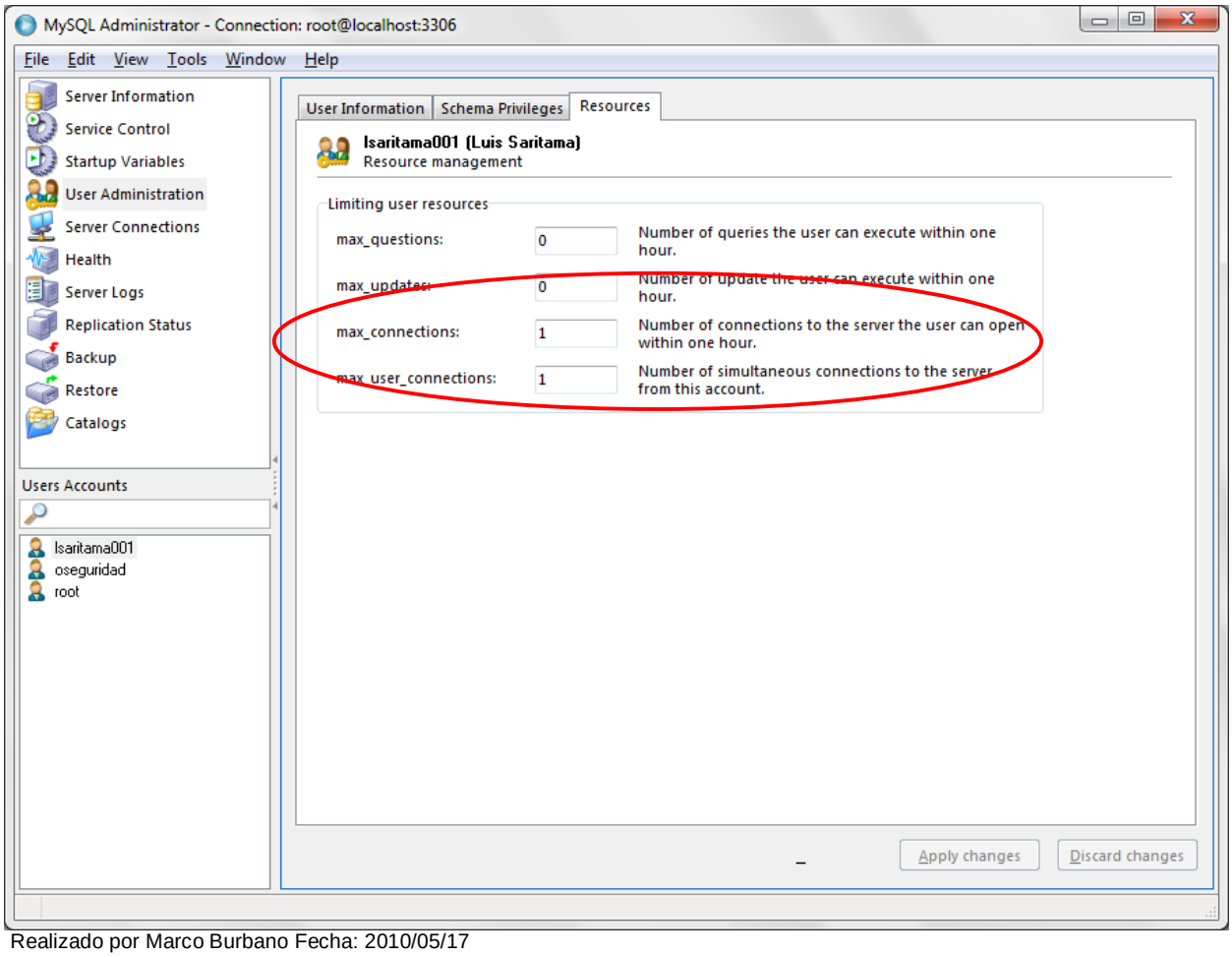
debe cuidar de no definir o quemar usuarios y contraseñas en los scripts a ejecutar.

6. **Limitar el número de sesiones concurrentes por cada usuario de preferencia permitir una sola sesión por cada usuario**

Implementación

MySql permite esta restricción, al momento de crear un usuario es posible configurar cuantas conexiones simultáneas se permite en la BDD.

Esta opción lo presenta el MYSQL Administrator en la administración de las cuentas de usuarios.



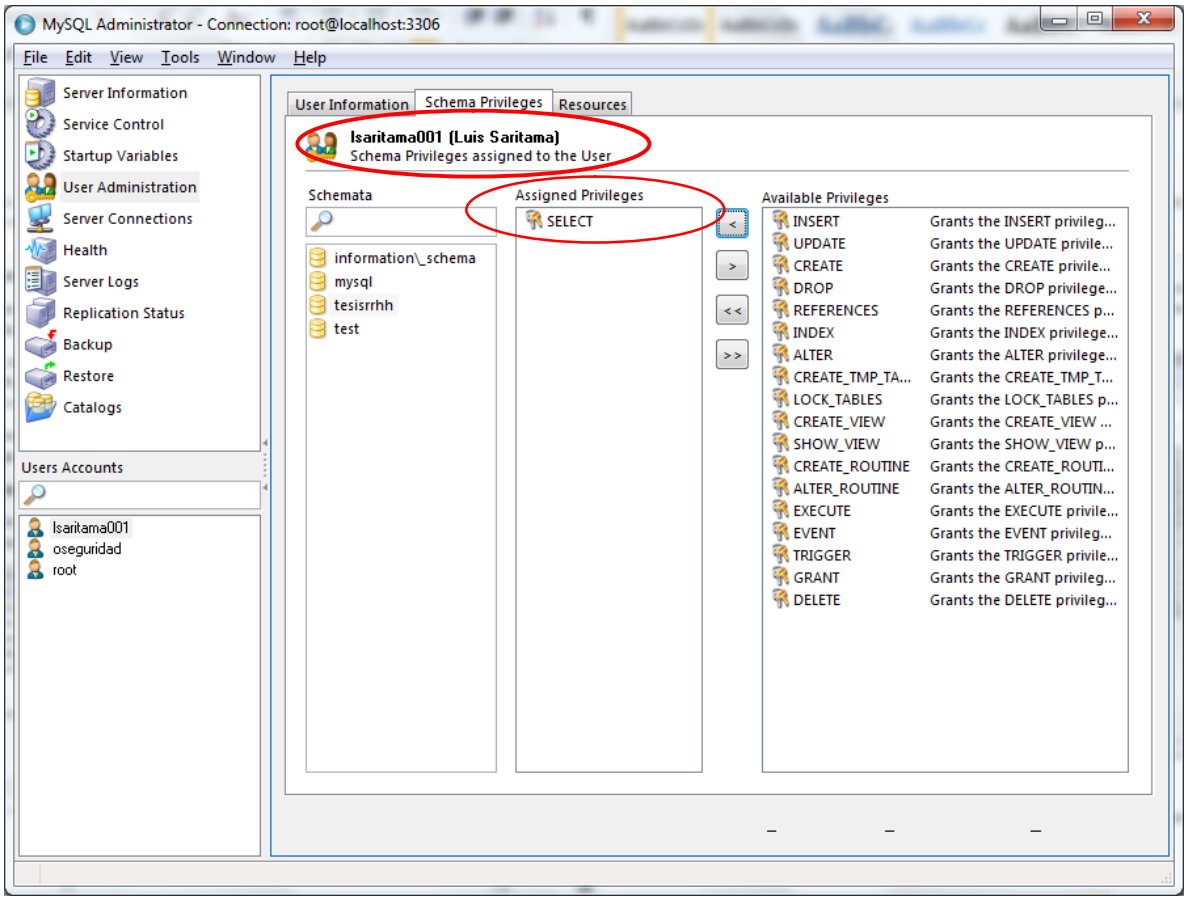
Realizado por Marco Burbano Fecha: 2010/05/17

7. La posibilidad de usar “GRANT OPTION” o “REVOKE” debe ser permitida solo a personal autorizado

Implementación

MySql permite esta restricción, esto se implementa en el momento de asignar a cada uno de los usuarios los respectivos privilegios de acuerdo a las funciones dentro de la organización.

Como se muestra esta opción solo se le otorga a los usuarios con la necesidad de generar y otorgar permisos en la BDD



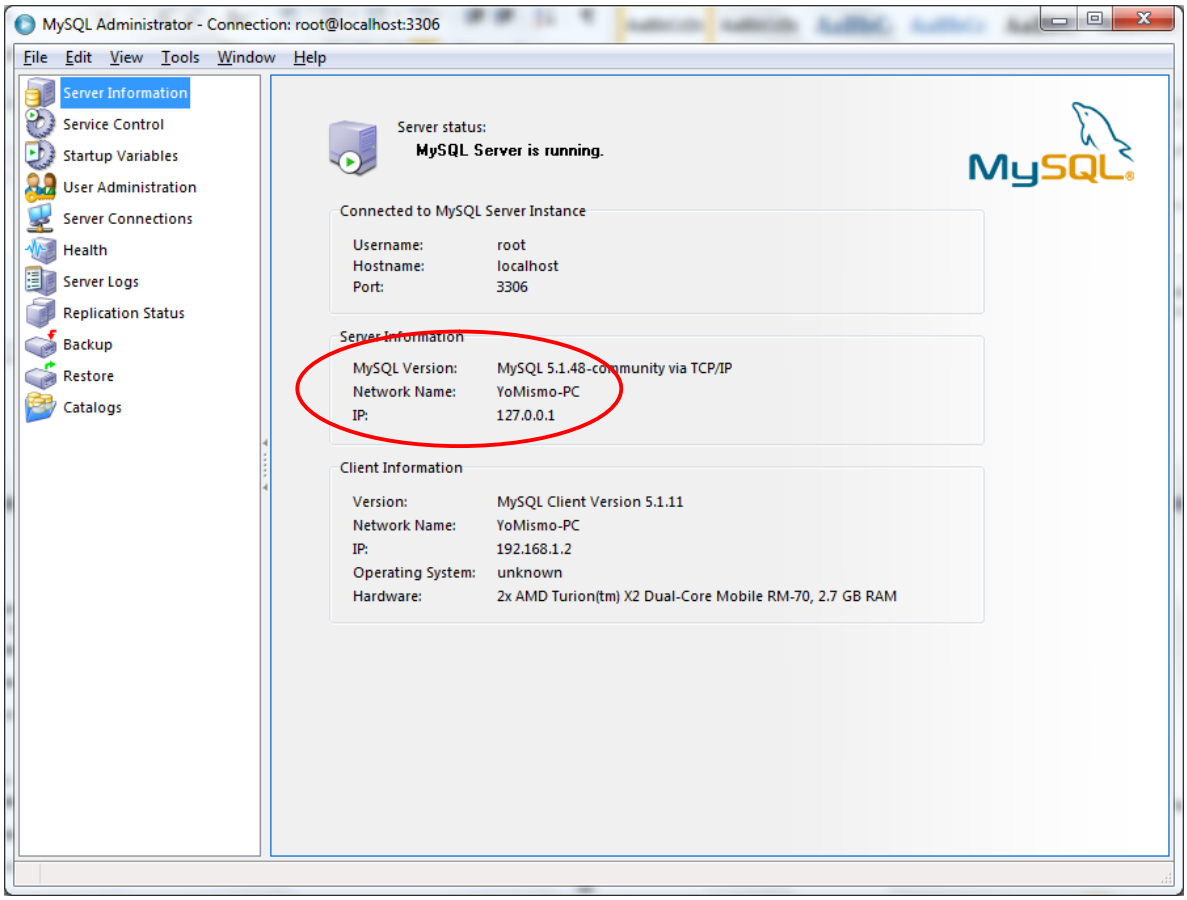
Realizado por Marco Burbano Fecha: 2010/05/17

En este caso como se trata de un usuario normal que tiene el cargo de asistente de RRHH no amerita mayores permisos.

8. Denegar la posibilidad de conectarse remotamente a la BDD más aún si los usuarios tienen permisos para modificar la información sensitiva

Implementación

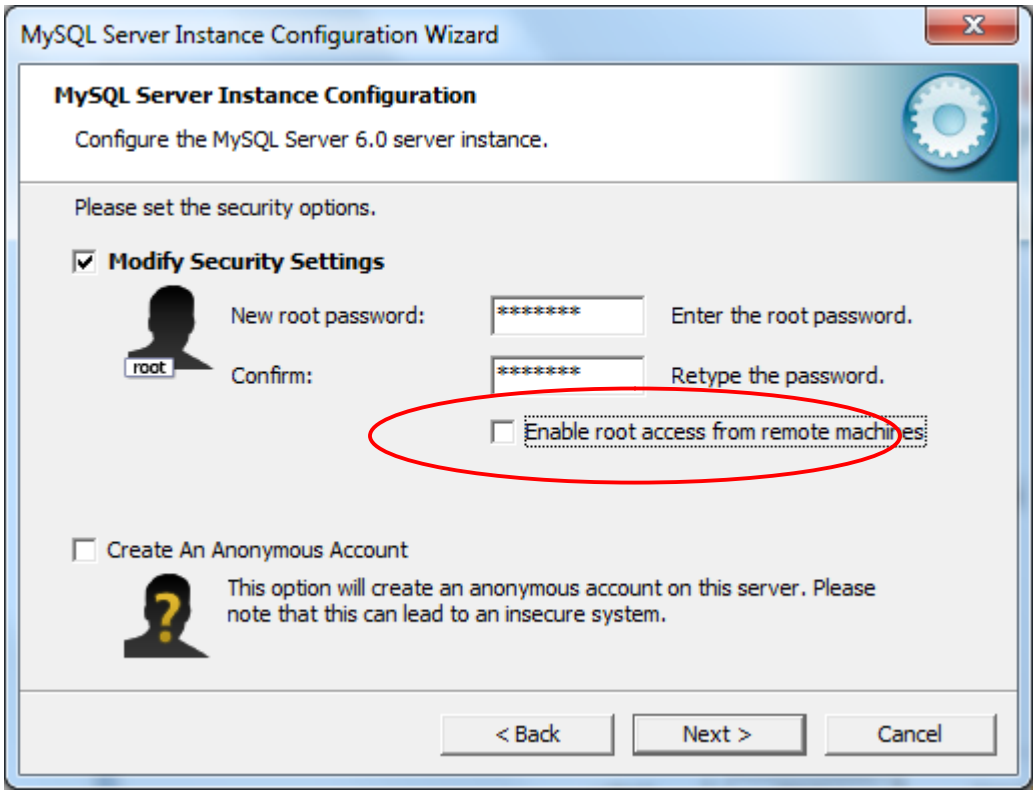
MYSQL permite esta restricción, mediante la configuración automática que maneja el motor de base de datos ya que no permite la conexión remota si no se cambia la IP del servidor que se defina automáticamente.



Realizado por Marco Burbano Fecha: 2010/05/17

Si fuese necesario el caso de permitir conexiones remotas debemos poner como parámetro en la IP la siguiente configuración “0.0.0.0” lo cual conlleva un riesgo el cual es necesario mitigar, una de las formas seria definir desde que host remoto se va a conectar y que usuario para esto existe una tabla de administración la cual nos permite definir estos parámetros, la tabla en mención es: tables_priv

Adicionalmente otra manera de configurar y restringir las conexiones remotas la encontramos en el momento de la instalación. En donde podemos escoger si se permite la conexión remota del usuario ROOT



Realizado por Marco Burbano Fecha: 2010/05/17

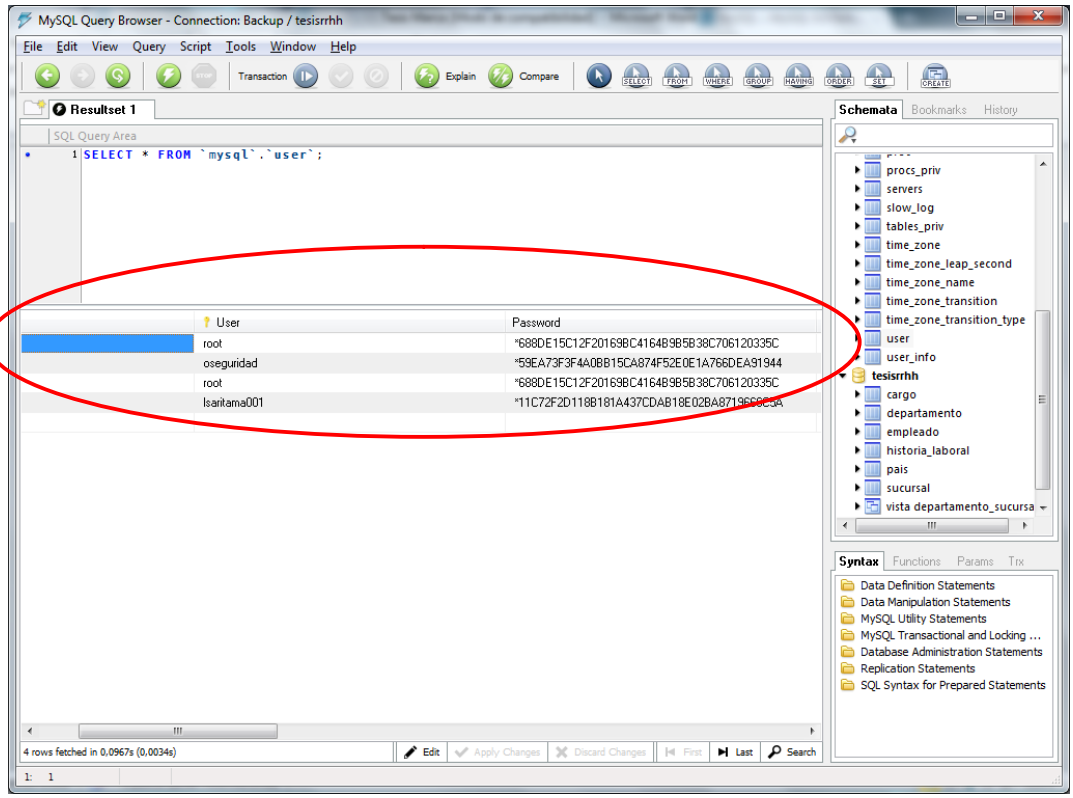
9. La información sensitiva como las contraseñas almacenadas en la base de datos debería ser encriptada. El no tener la información primordial encriptada aumenta el riesgo de accesos no autorizados.

Implementación

"MySQL encripta contraseñas usando su propio algoritmo. Esta encriptación es diferente de la usada durante el proceso de logueo de Unix. La encriptación de contraseña es la misma que la implementada en la función PASSWORD(). La

encriptación de contraseñas Unix es la misma que la implementada por la función SQLENCRYPT ().”[17]

Aquí podemos ver que efectivamente MYSQL utiliza la función antes mencionada.



Realizado por Marco Burbano Fecha: 2010/05/17

10. Activar los registros de auditoría con la finalidad de llevar un control sobre las actividades realizadas en la BDD

Implementación

Se lo debería implementar mediante un Trigger o un aplicativo, el cual haya sido configurado de manera adecuada, ya que realizar Triggers de manera

[17]Obtenido de: http://dev.mysql.com/doc/refman/5.0/es/user-names.html Octubre 2010

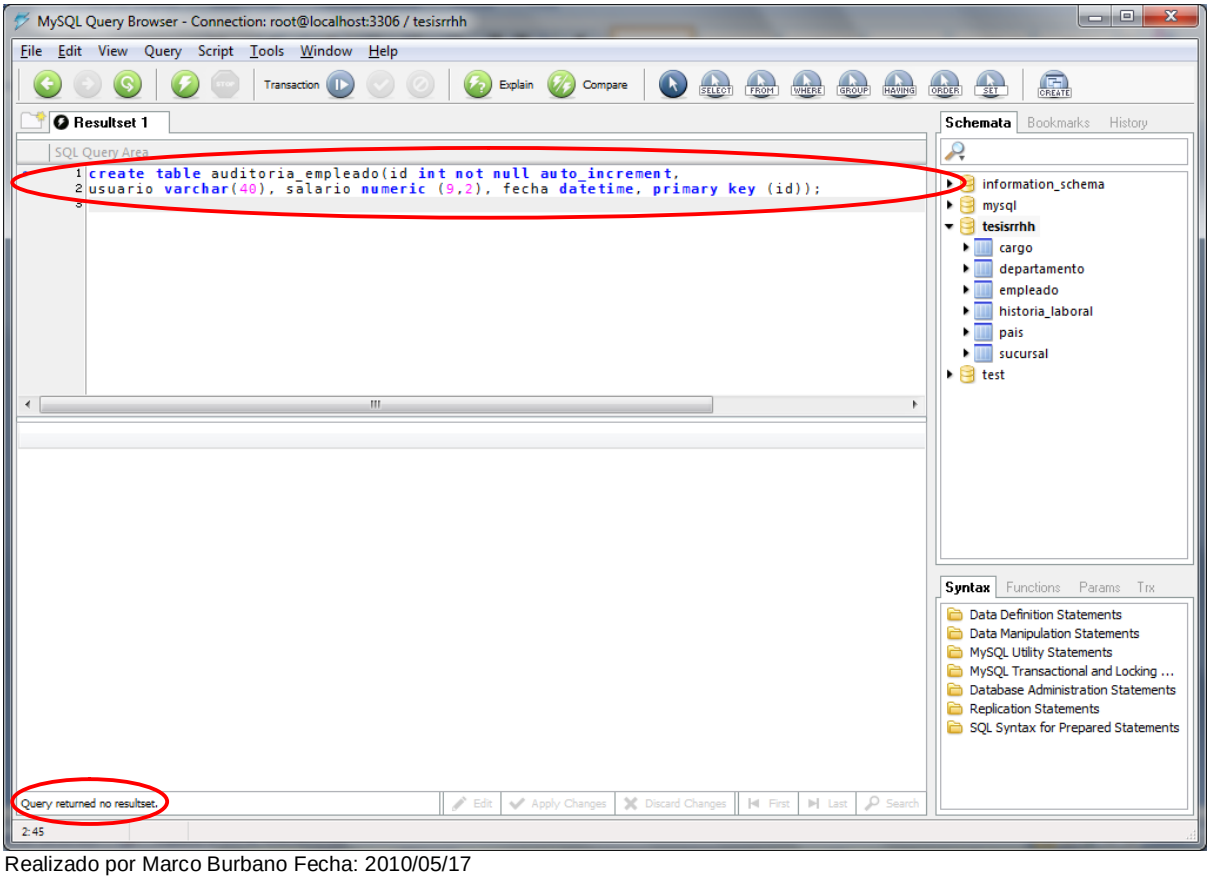
Para este caso implementaremos un trigger el cual nos permita auditar las actualizaciones a los salarios que se ejecuten en la tabla empleados.

Para esto crearemos una tabla de Auditoria llamada “auditoria_empleado”, en la cual se registraran las modificaciones a los salarios.

/*Tabla Auditoria*/

create table auditoria_empleado(id int not null auto_increment,

usuario varchar(40), salario numeric (9,2), fecha datetime, primary key (id));



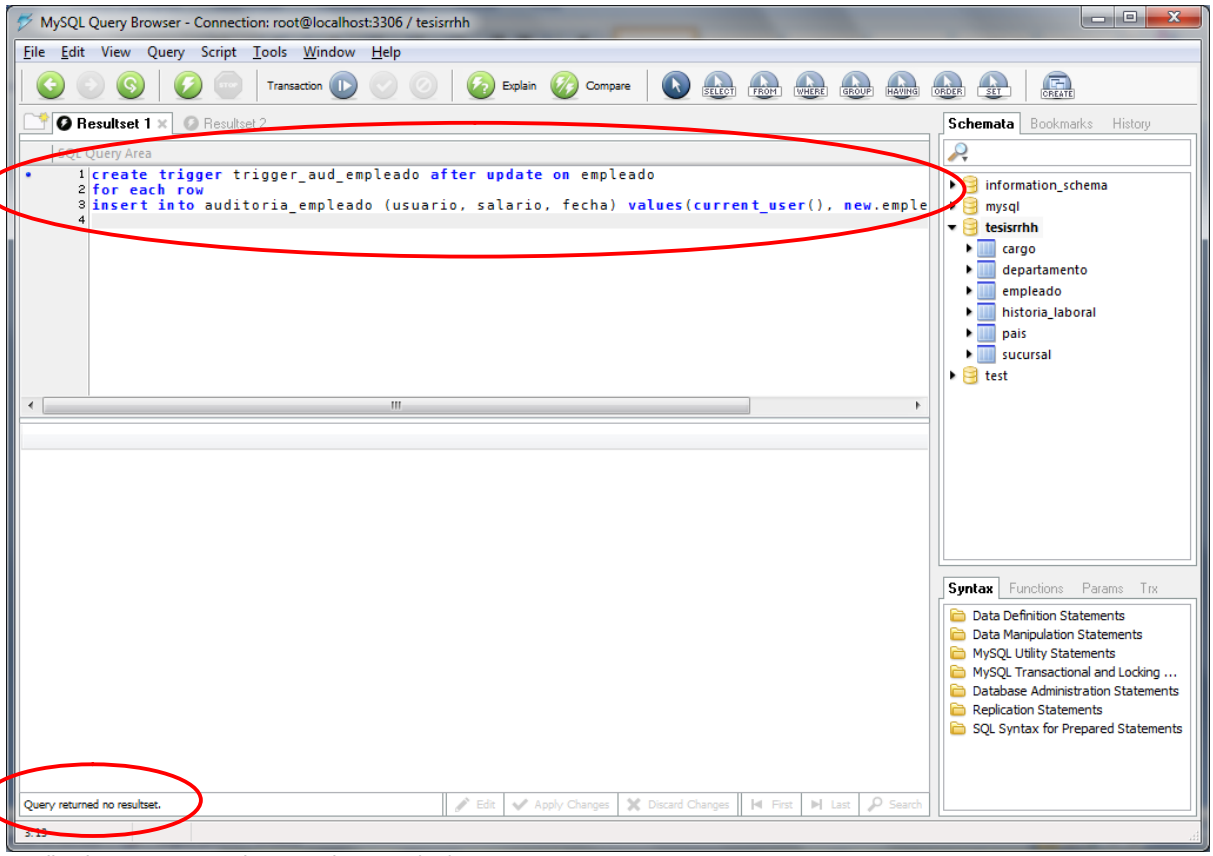
Realizado por Marco Burbano Fecha: 2010/05/17

/*Trigger Auditoria*/

create trigger trigger_aud_empleado after update on empleado

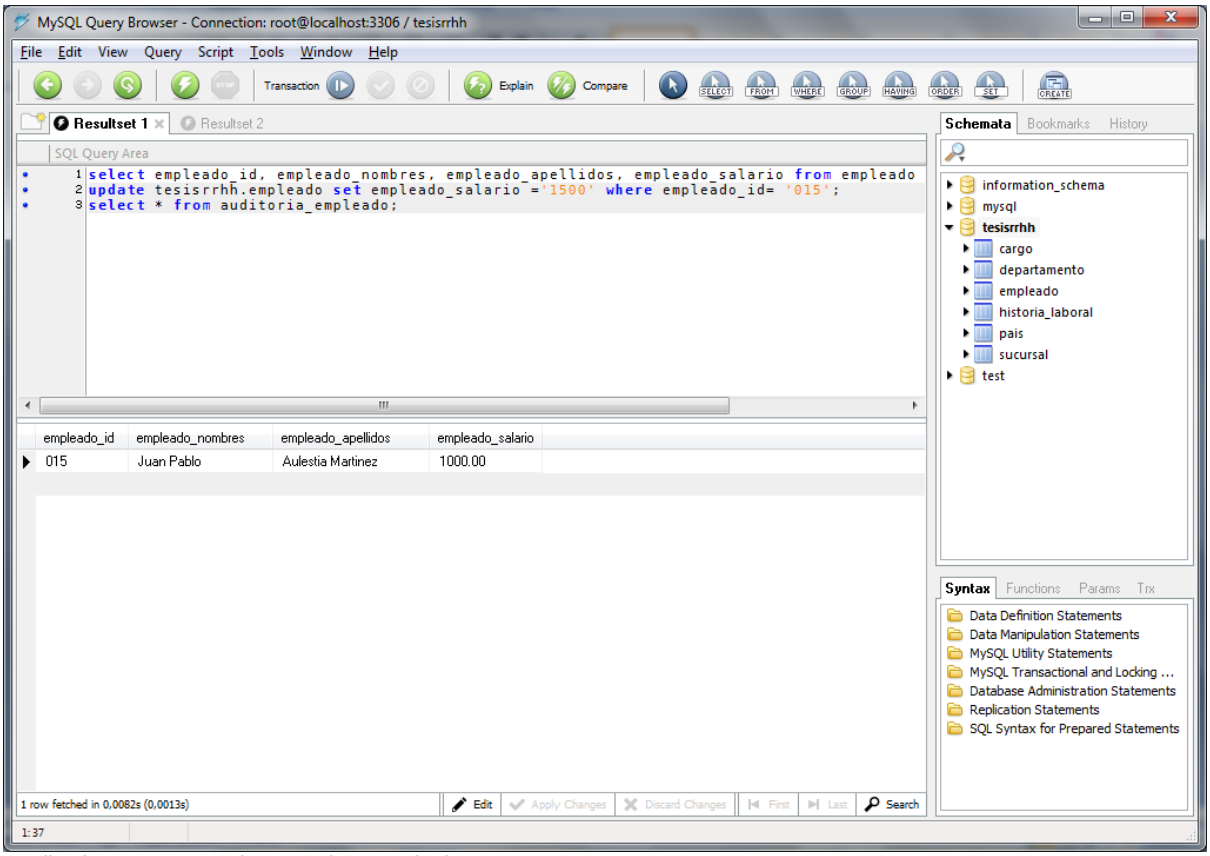
for each row

insert into auditoria_empleado (usuario, salario, fecha) values(current_user(), new.empleado_salario, now());



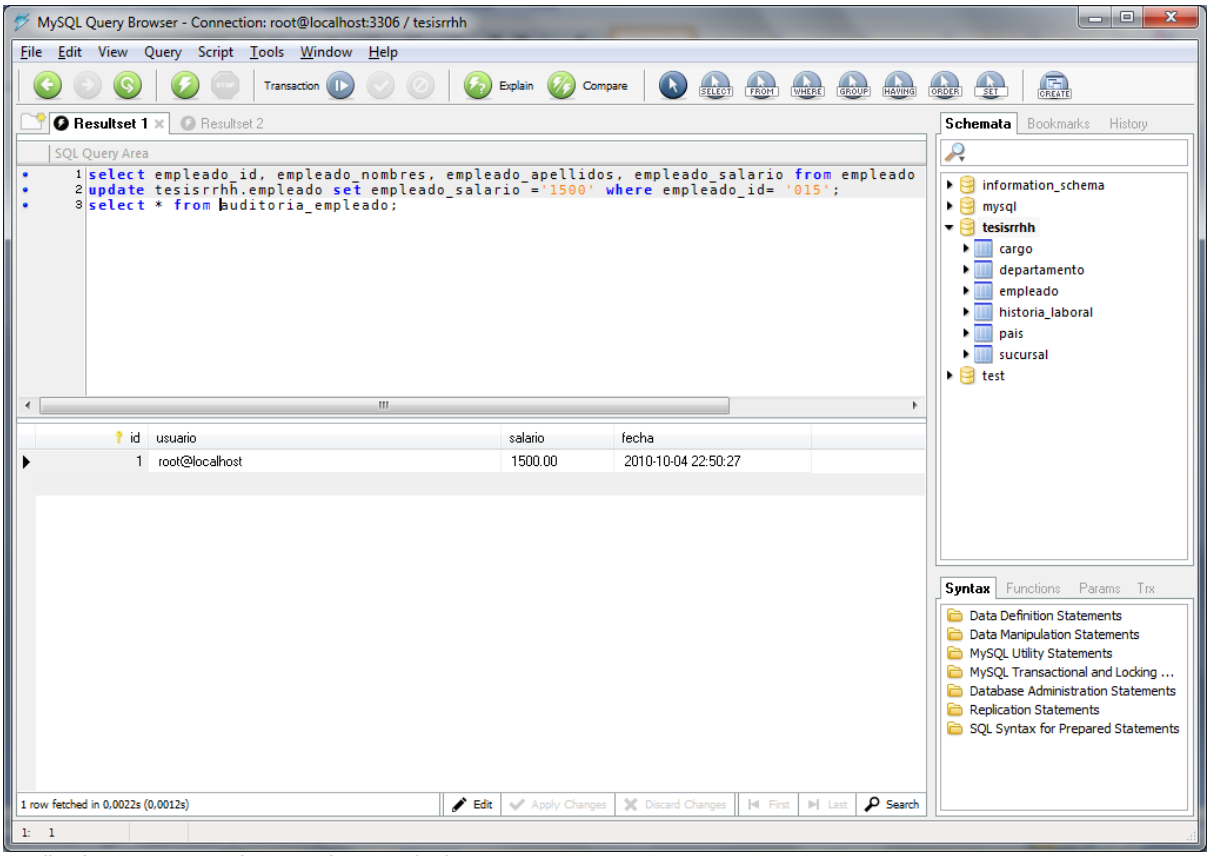
Realizado por Marco Burbano Fecha: 2010/05/17

Revisamos si el trigger funciona correctamente, verificamos el usuario que vamos a editar.



Realizado por Marco Burbano Fecha: 2010/05/17

Realizamos el update y verificamos si se guardo el registro en la tabla de auditoría que creamos previamente.



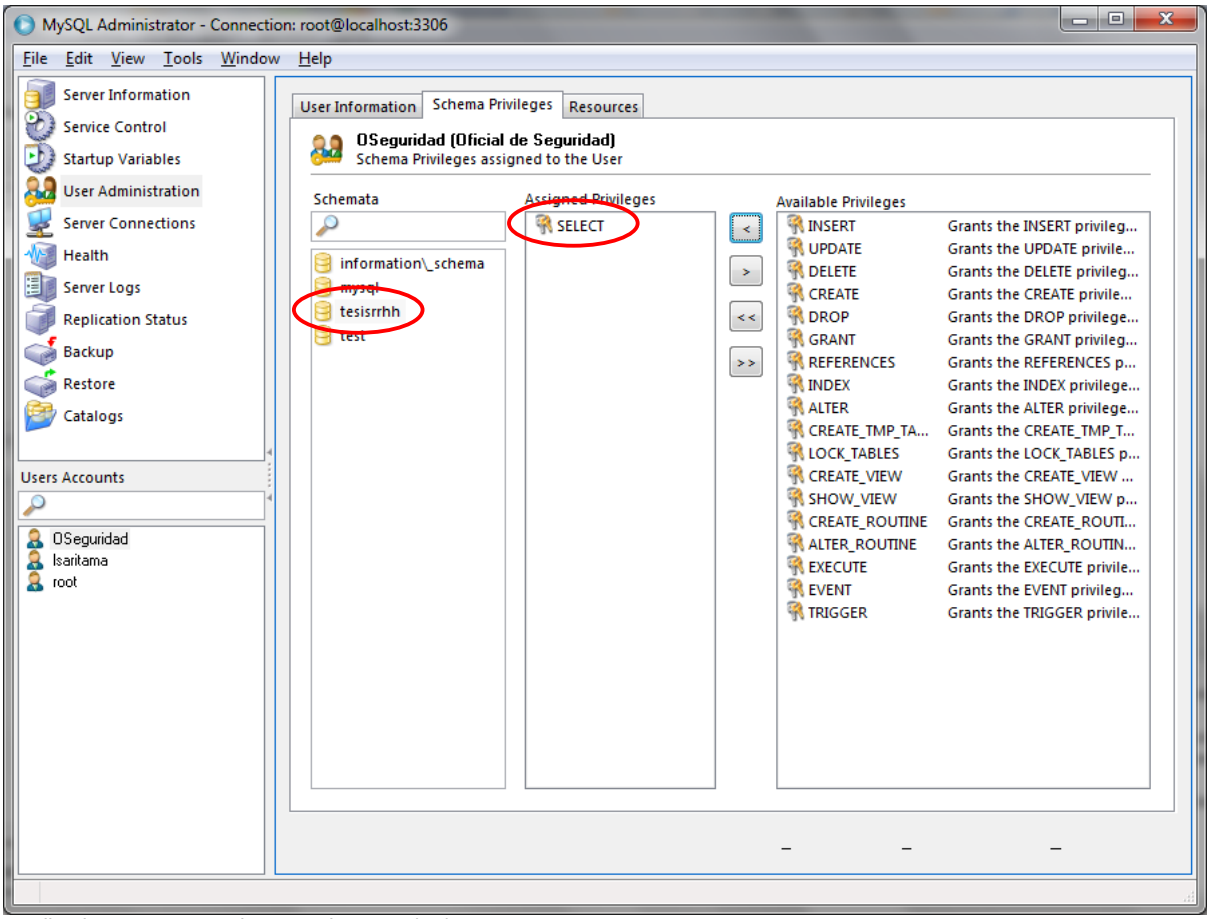
Realizado por Marco Burbano Fecha: 2010/05/17

11. El administrador de seguridad (no el DBA) debe supervisar la configuración de auditoría de base de datos. Sólo el administrador de seguridad debe tener acceso a los registros de auditoría.

Implementación

Se lo implementa mediante roles y permisos, efectuando una adecuada implementación de los roles generando una correcta segregación de funciones. Y acceso a solo los archivos de auditoria

En este caso solo brindaremos privilegios de visualización de las tablas del esquema tesisrrhh



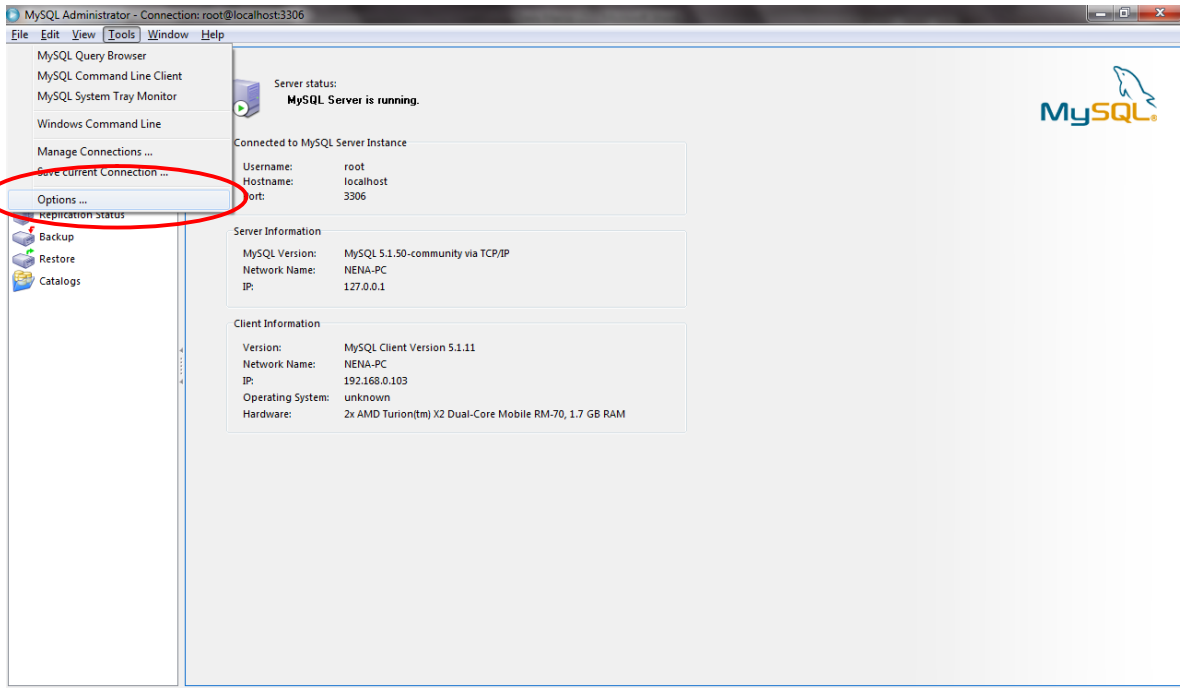
Realizado por Marco Burbano Fecha: 2010/05/17

12. Asegurar que los controles de Respaldo y Restauración de la Base de Datos garantiza la disponibilidad de los datos los cuales se pueden recuperar por completo

Implementación

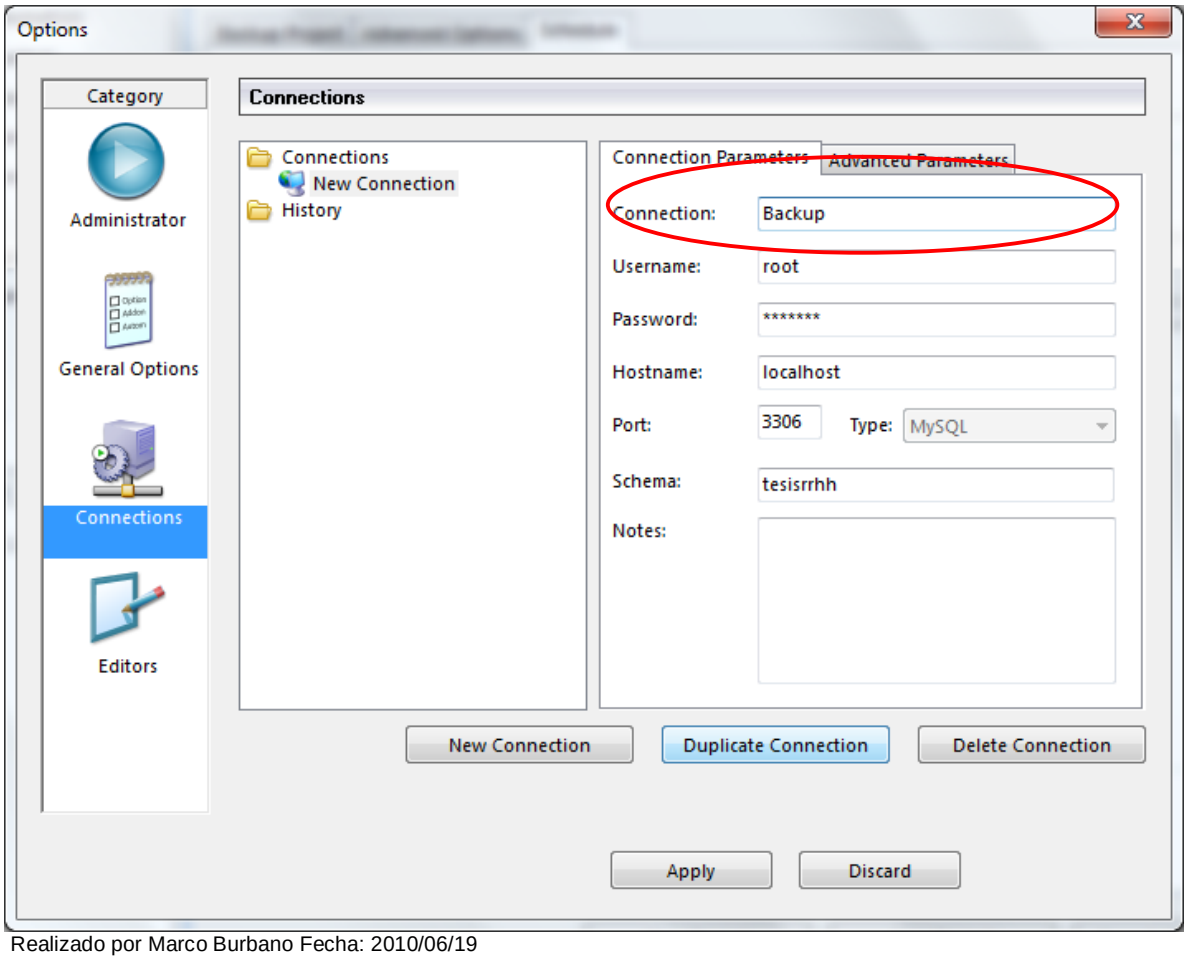
MySql permite ejecutar respaldos la revisión va por cuenta del departamento de IT, para poder realizarlo es necesario definir una conexión que se ejecute automáticamente para poder realizar el backup los días que se especifiquen.

Para configurar la ejecución del Backup ingresamos al MySQL Administrator en donde primero debemos configurar una conexión con la cual se ejecutara el procedimiento de respaldo. Nos dirigimos a la pestaña TOOLS y seleccionamos la opción de OPTIONS



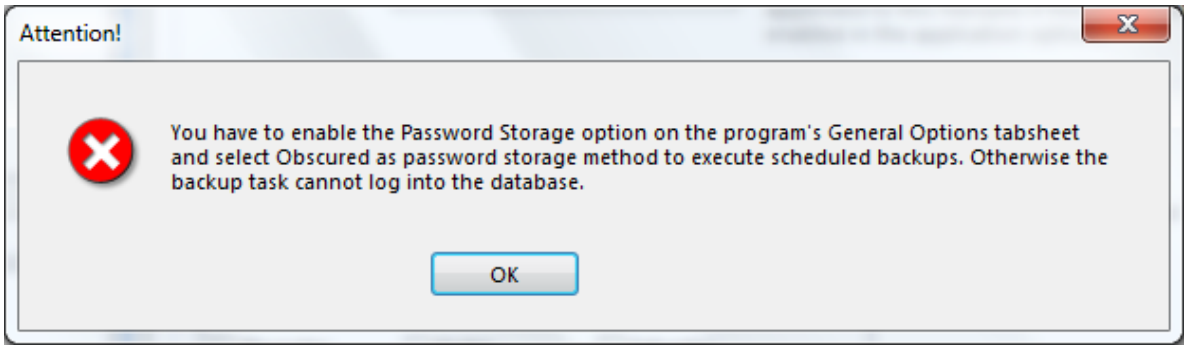
Realizado por Marco Burbano Fecha: 2010/06/19

Después seleccionamos la opción de Connections en donde configuramos los parámetros iníciales de la nueva conexión de Backup. Además necesitamos que sea un usuario y contraseña con privilegios ya que de lo contrario no podrá ejecutar el backup.

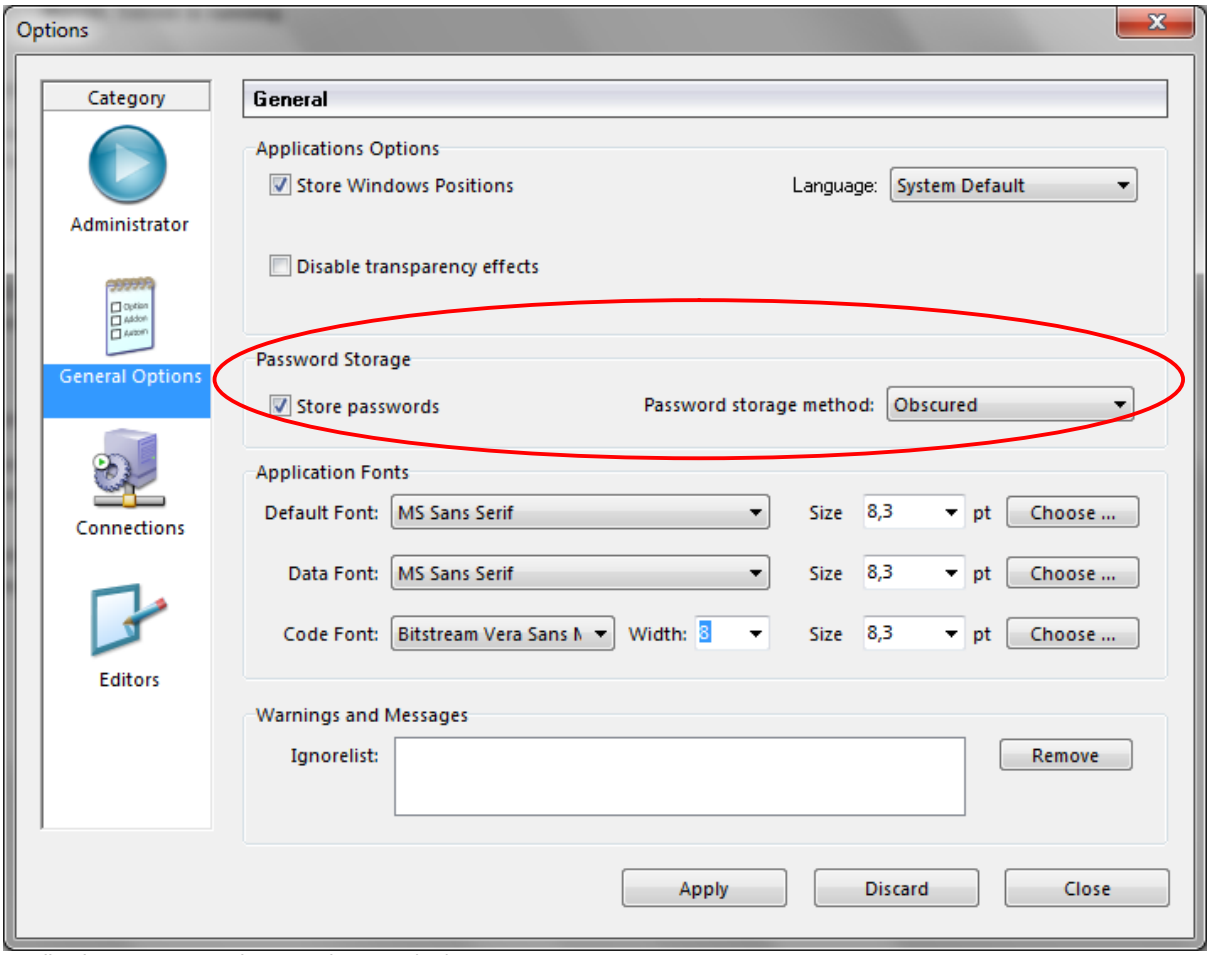


Realizado por Marco Burbano Fecha: 2010/06/19

Posteriormente nos aparecerá un mensaje pidiendo que se pueda almacenar y ejecutar la conexión mediante el uso de la contraseña con la opción “Obscure” en PasswordEstorage.



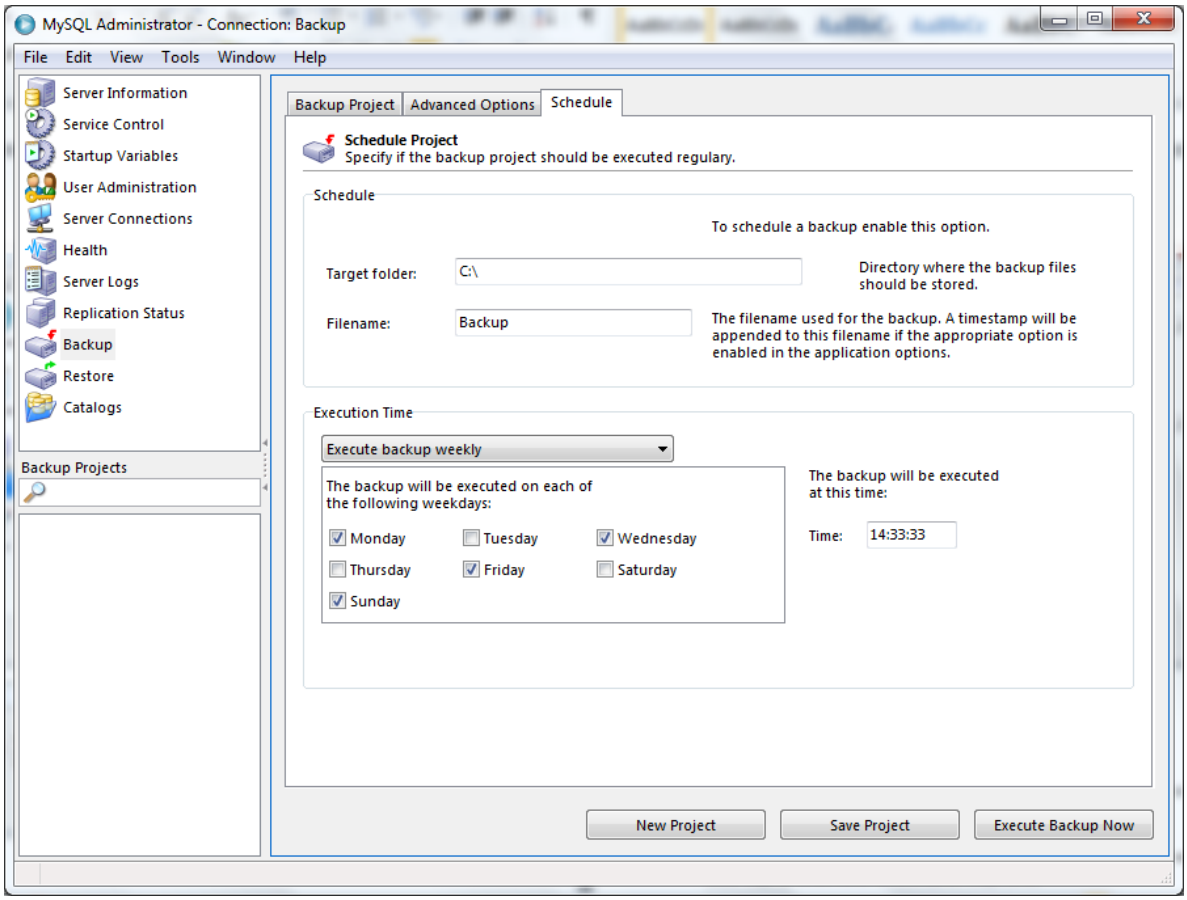
Realizado por Marco Burbano Fecha: 2010/06/19



Realizado por Marco Burbano Fecha: 2010/06/19

Finalmente es posible configurar la realización de backups de manera automática, de acuerdo a las necesidades y políticas establecidas en la organización.

Además cabe mencionar que después de obtener los backups estos deben ser probados con la finalidad de asegurar el correcto funcionamiento, para posteriormente llevarlos a un lugar externo a la organización. Mediante la definición de un lugar seguro donde almacenarlas y garantizar que en caso de necesitarlas estas listas y completas para utilizarlas. Por ejemplo la contratación de las bóvedas bancarias es uno de los lugares más idóneos para el almacenamiento de los backups de información.



Realizado por Marco Burbano Fecha: 2010/06/19

13. Realizar una correcta asignación de privilegios a cada uno de los usuarios

Implementación

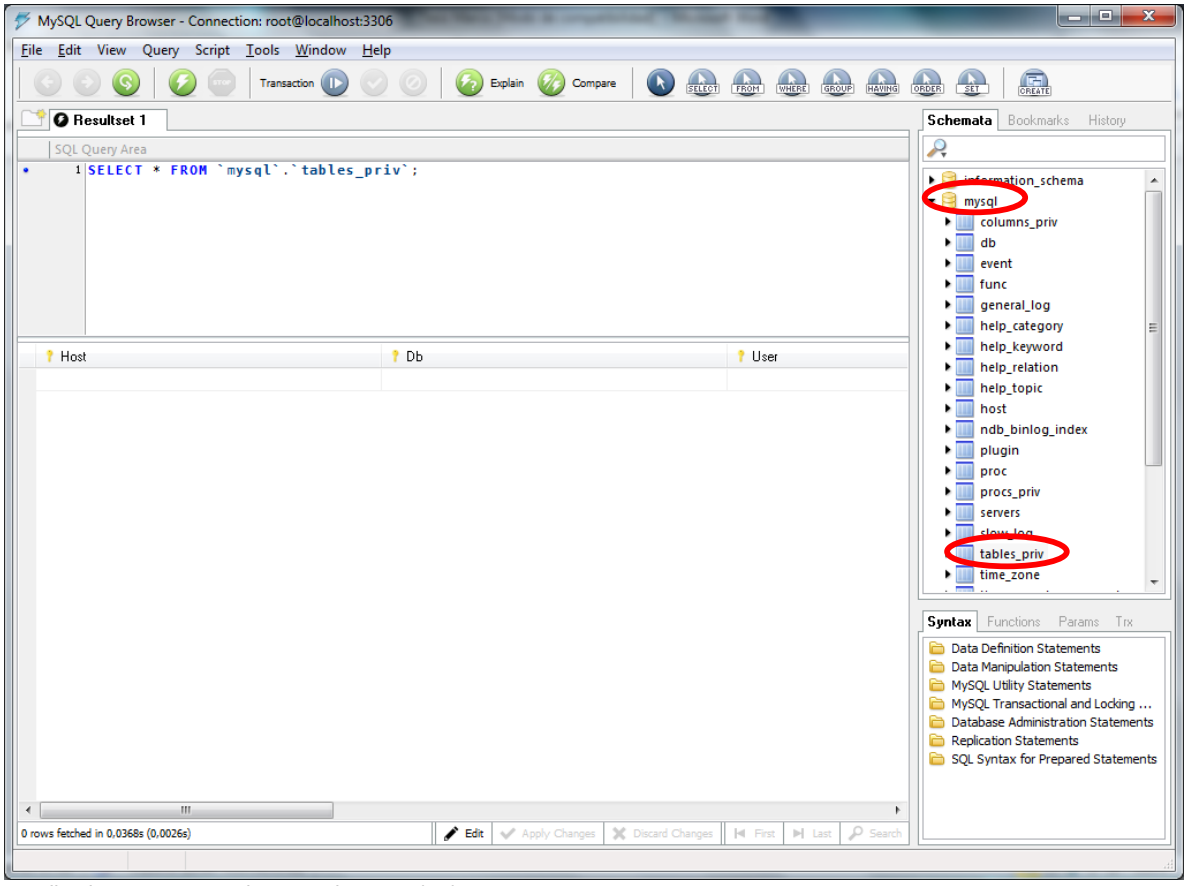
Implementar procedimientos rigurosos para la asignación de perfiles y roles en la BDD, cada organización debe definir procedimientos y políticas que debe seguirse con la finalidad de minimizar el riesgo de asignar permisos incorrectos y que esto impacte directamente la información que se almacena en el motor de BDD.

14. Todos los identificadores de usuario debe ser único e identificar a un único usuario y debe respetar la norma de denominación corporativa.

Implementación

Definir políticas para el acceso a la BDD, eso lo debe establecer el gerente de TI con una delegación entre los cuales debe estar el gerente general con la finalidad de darle importancia a las políticas que se establecen.

Adicionalmente MYSQL puede restringir y otorgar accesos a nivel de tabla y campo, esto mediante la Base de administración “mysql”.



Realizado por Marco Burbano Fecha: 2010/06/19

15. Restringir el uso de líneas de comando a usuarios autorizados

Implementación

Una buena práctica es restringir la ejecución de comandos que afecten directamente a la BDD (Querys y/o Comandos de administración), a esto se lo puede restringir de varias maneras, una de ellas es la siguiente.

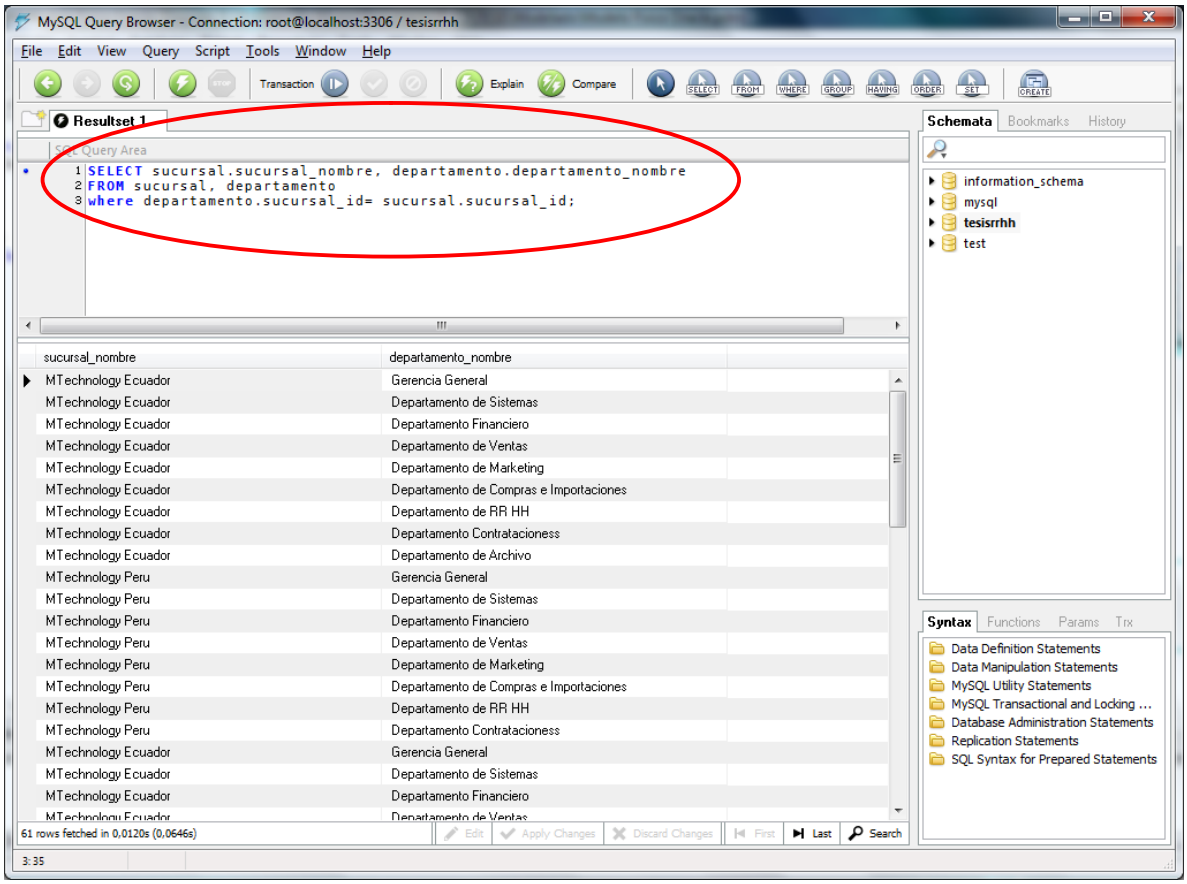
Se debe limitar la posibilidad de tener instalados interfaces de conexión directa a la Base de Datos. Como por ejemplo limitar la posibilidad de instalar MySQL Front, MyQuery Browser o cualquier consola de ejecución de consultas o comandos.

16. Aplicación de vistas con la finalidad de restringir la información a usuarios finales

Implementación

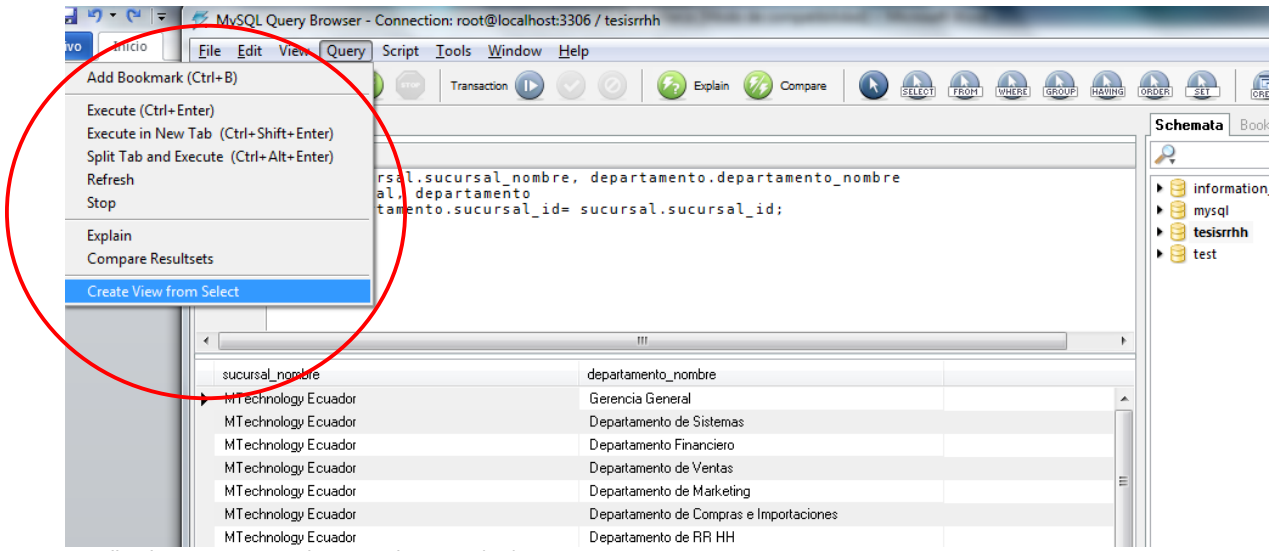
MYSQL permite la realización de Vistas.

Definimos la consulta que se va a transformar en una vista esto se lo puede realizar mediante líneas de código (Querys) o con la ayuda de la interfaz gráfica proporcionada por MYSQL.



Realizado por Marco Burbano Fecha: 2010/06/19

En esta ocasión crearemos la vista mediante la funcionalidad de MYSQL



Realizado por Marco Burbano Fecha: 2010/08/02

17. Aplicar integridad referencial

Implementación[18]

Implementado en MySql ya que este motor de base de datos actualmente posee varios tipos de tablas gestionadas por diferentes motores, estas tablas son:

ISAM: es el formato de almacenaje más antiguo, y posiblemente podría desaparecer en futuras versiones. Presentaba limitaciones importantes como la no exportación de ficheros entre maquinas de distintas arquitecturas o que no podía usar mayores de 4 Gigabytes.

MYISAM: es el tipo de tabla por defecto en MySQL desde la versión 3.23. Optimizada para sistemas operativos de 64 bits, permite ficheros de tamaños mayores que las ISAM. Los datos se almacenan en un formato independiente, lo que permite pasar tablas entre distintas plataformas. Los índices se almacenan en un archivo con la extensión ".MYI" y los datos en otro archivo con extensión ".MYD". Ofrece la posibilidad de indexar campos BLOB y TEXT. Además este tipo de tablas soportan el tipo de dato VARCHAR.

Un inconveniente es que las tablas pueden llegar a corromperse, almacenando datos incorrectos. Esto puede ser causado por:

- El proceso mysqld haya sido eliminado en el transcurso de una escritura.
- Problemas de hardware.

[18] http://www.webtaller.com/construccion/lenguajes/mysql/lecciones/tipos_tablas_usadas_mysql.php Obtenido el 29 de octubre de 2010

- Una caída del sistema durante su utilización.

- Un gusano en el código Mysql o MyISAM.

INNODB: InnoDB provee a MySQL con el soporte para trabajar con transacciones, además de hacer un mejor bloqueo de registros para las instrucciones SELECT muy parecido al usado por Oracle, con lo que incrementa el rendimiento y la concurrencia en ambientes multiusuario, por otro lado, InnoDB es el único formato que tiene MySQL para soportar llaves foráneas (FOREING KEY). Además de todo lo comentado, InnoDB ofrece unos rendimientos superiores a la anterior tecnología de tablas de MySQL (MyISAM).

InnoDB es un motor de bases de datos muy completo que ha sido integrado dentro de MySQL.

Otras de sus características son:

- Recuperación automática ante fallas. Si MySQL se da de baja de una forma anormal, InnoDB automáticamente completará las transacciones que quedaron incompletas.

- Integridad referencial. Ahora se pueden definir llaves foráneas entre tablas InnoDB relacionadas para asegurarse de que un registro no puede ser eliminado de una tabla si aún está siendo referenciado por otra tabla.

- Bloqueo a nivel de filas. Al usar tablas MyISAM, y tener consultas muy grandes que requieren de mucho tiempo, simplemente no se podían ejecutar más

consultas hasta que terminarán las consultas que estaban en ejecución. En cambio, las tablas InnoDB usan bloqueo a nivel de filas para mejorar de manera impresionante el rendimiento.

- SELECTs sin bloqueo. El motor InnoDB usa una técnica conocida como multi-versioning (similar a PostgreSQL) que elimina la necesidad de hacer bloqueos en consultas SELECT muy simples. Ya no será necesario molestarse porque una simple consulta de sólo lectura está siendo bloqueada por otra consulta que está haciendo cambios en una misma tabla.

HEAP: Tablas en memoria. Son temporales y desaparecen cuando el servidor se cierra, a diferencia de una tabla TEMPORARY, que solo puede ser accedida por el usuario que la crea, una tabla HEAP puede ser utilizada por diversos usuarios. No soportan columnas de autoincremento ni que haya valores nulos en los índices. Los datos son almacenados en pequeños bloques.

BDB: Base de datos Berkeley. TST(Transactions safe tables). Solo en MySQL

MAX: Este tipo de tablas permite la realización de transacciones (a partir de la versión 3.23.34), por lo que es posible la recuperación de datos (COMMIT y ROLLBACK). Estas tablas necesitan de una clave primaria en cada tabla, que ha de crear el administrador o de lo contrario Mysql creará una oculta. Otra de sus características es que pueden ser bloqueadas con el comando LOCK. Estas tablas son almacenadas en archivos ".DB".

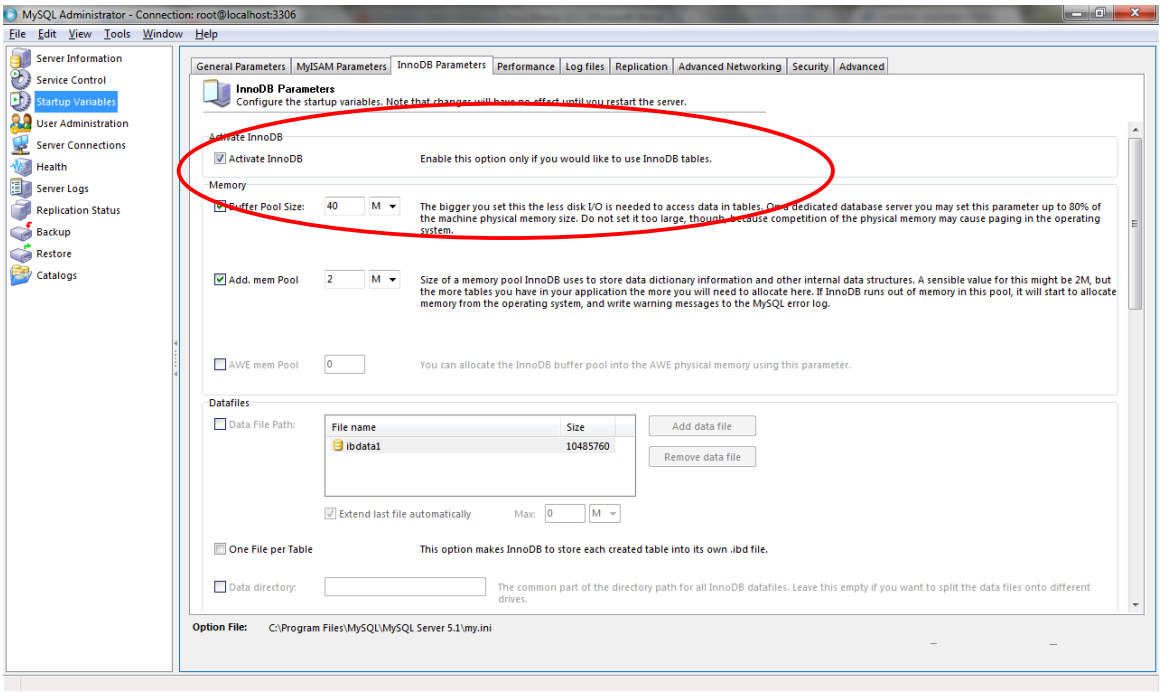
TST: *'Transactions safe tables'*, o tablas para transacciones seguras. Son menos rápidas y ocupan más memoria, pero a cambio ofrecen mayor seguridad frente a fallos durante la consulta. Las tablas TST permiten ir introduciendo consultas y finalizar con un COMMIT (que las ejecuta) o ROLLBACK (que ignora los cambios). Disponibles a partir de la versión 4 de MySQL.

Después de esta información queda claro que para poder aplicar integridad referencial en una basa de datos con MySQL es necesario utilizar las tablas INNODB con lo cual podemos utilizar las Foreing Keys así como también la posibilidad de ejecutar COMMIT Y ROLLBACK.

Para configurar en qué tipo de tablas queremos almacenar la información podemos ejecutar el siguiente query.

```
ALTER TABLE nombreDeLaTabla ENGINE=INNODB;
```

Para configurar ciertos parámetros de este tipo de tabla podemos ir a la Administración del MySQL Administrator a la Pestaña STARTUP VARIABLES y posteriormente al Menú de InnoDB Parameters en donde podemos seleccionar la utilización de estas tablas así como también de los diferentes parámetros que cada organización necesite.



Realizado por Marco Burbano Fecha: 2010/08/02

Con lo cual se obtiene los beneficios de poder implementar foreing keys en las tablas de nuestra base de datos, así como también la posibilidad de utilizar commits y rollbacks.

18. Utilización de clúster para mejorar la disponibilidad

Implementación

MySQL tiene una herramienta adicional que proporciona un soporte para la configuración y mantenimiento de clusters.

“MySQLCluster es una versión de alta disponibilidad, alta redundancia de MySQL adaptada para el entorno de computación distribuida. Usa el motor de

almacenamiento *NDB Cluster* para permitir la ejecución de varios servidores MySQL en un cluster. Este motor de almacenamiento está disponible en las distribuciones binarias de MySQL 5.0 y en los RPMs compatibles con las distribuciones Linux más modernas. (Tenga en cuenta que tanto los RPMs mysql-server como mysql-max deben instalarse para tener la capacidad de MySQLCluster .)"[19]

19. Utilización del servicio de un Banco de Datos

Implementación

La utilización del servicio de Banco de Datos no depende directamente del Motor de Base de Datos puesto, que el servicio funciona y lo administra directamente un tercero, quien es el responsable de garantizar la disponibilidad, integridad y seguridad de la información que la organización deposite.

20. Utilización de redundancia

Implementación

Si bien MYSQL admite la ejecución de replicación, es necesario mencionar que existen varias maneras de implementar una redundancia controlada, Por ejemplo:

Se puede definir que se tiene un servidor Beta o Esclavo en el cual se tendrá la réplica controlada. Además será necesario realizar un backup con la finalidad de

[19] http://dev.mysql.com/doc/refman/5.0/es/ndbcluster.html Obtenido el 29 de octubre de 2010

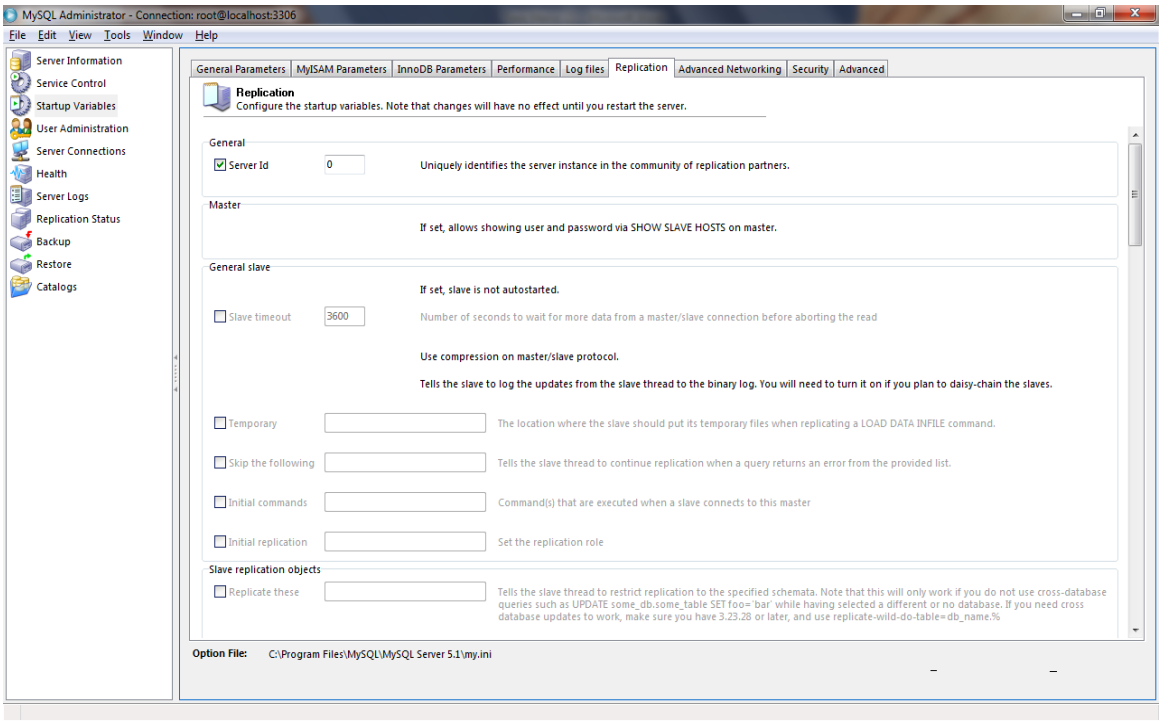
subir al servidor Beta el respaldo obtenido, así verificar el correcto funcionamiento de la BDD, cabe mencionar que dependiendo de la frecuencia con la que se realice el backup, la disponibilidad aumentara o disminuirá proporcionalmente.

Otra forma de realizar redundancia a nivel de BDD es mediante la implementación de un sistema que permita realizar la replicación en línea de cada una de las transacciones que se realicen en la base de datos.

Finalmente este procedimiento se lo puede realizar mediante procedimientos batch los cuales se los ejecuta de manera automática, esto dependerá de las instrucciones que ingresemos en la programación. Lo que sí es importante recordar es que se debe realizar una revisión con la finalidad de que los datos replicados son los correctos, garantizando la integridad y disponibilidad de la información.

MySQL permite la replicación mediante la configuración de variables ingresando a la Consola de administración.

Nos dirigimos a Startup Variables, posteriormente escogemos la pestaña de REPLICATION en donde configuramos la manera como queremos configurar la replicación.



Realizado por Marco Burbano Fecha: 2010/08/23

3.3Implementación en una Base de Datos de Arquitectura Comercial

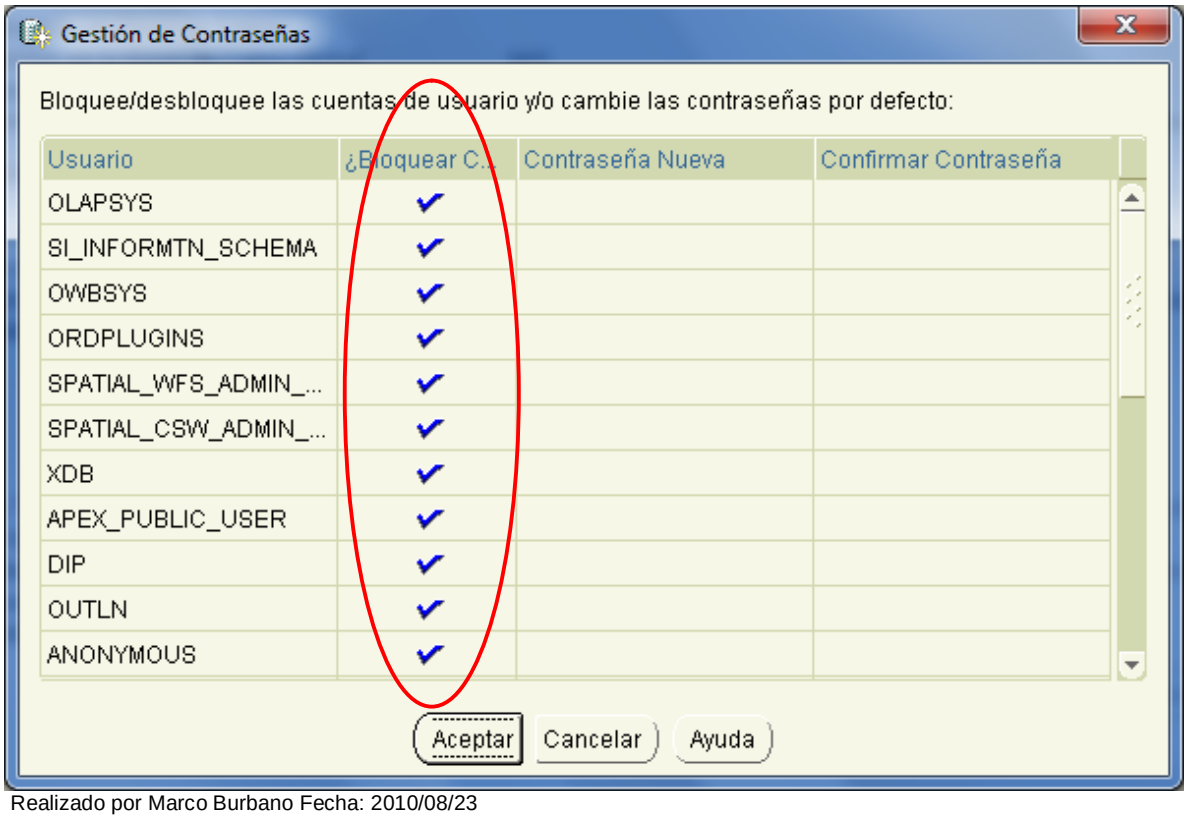
1. **Se debe mantener bloqueados a los usuarios creados automáticamente por la BDD y en el mejor de los casos se los debe Borrar con la finalidad de mantener la seguridad.**

Implementación

ORACLE cuando se la instala mediante líneas de código en cualquiera que sea el sistema operativo sobre el cual se montará la base de datos, estos scripts pueden ser modificados de manera que la configuración inicial de la BDD puede dejar a la seguridad por debajo de los niveles óptimos para el correcto funcionamiento de la BDD. Es por esto que se recomienda que la configuración de estos scripts limite la activación de usuarios que no son necesarios para la configuración inicial de la BDD.

Por otro lado si instalamos Oracle mediante una herramienta gráfica proporcionadas por SUN (OUI)[20], éstas por defecto generan bloqueados la mayoría de los usuarios por defecto del Motor de Base de Datos (p.e. Scott, hr,anonymous, etc.). Con lo cual mantenemos la seguridad e integridad de la BDD que estamos configurando. Limitando la posibilidad de acceso no autorizado a la información que se almacenará en la BDD

[20] OUI: Oracle Universal Instaler



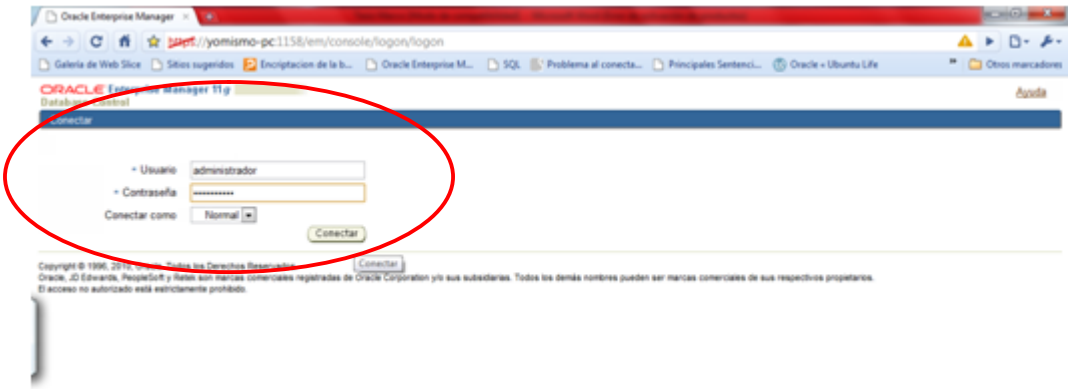
Realizado por Marco Burbano Fecha: 2010/08/23

2. **Se debe realizar segregación de funciones en el SO y La BDD con la finalidad de no colocar en una sola persona la completa seguridad del ambiente en el que se encuentra la BDD**

Implementación

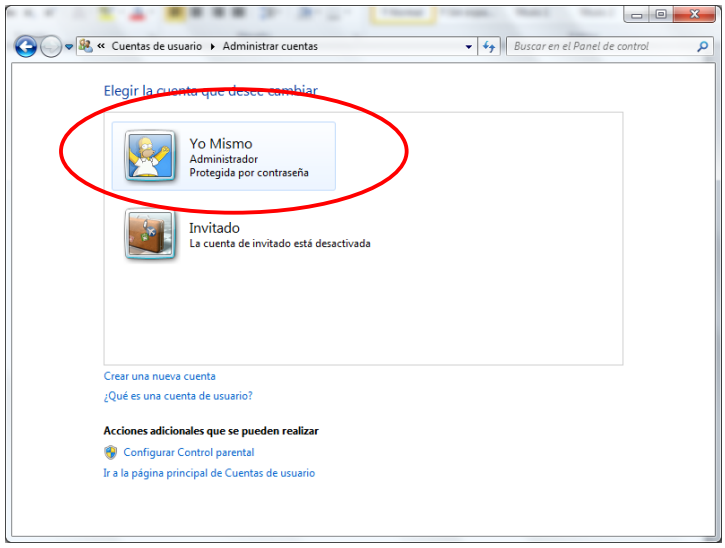
Se la realiza evitando una autenticación en la BDD mediante la autenticación del SO y para una mayor seguridad las claves de mayor poder en La BDD y SO deben pertenecer a usuarios diferentes.

Autenticación en la Base de Datos



Realizado por Marco Burbano Fecha: 2010/08/23

Autenticación en el SO mediante una contraseña.



Realizado por Marco Burbano Fecha: 2010/08/23

3. **Mediante aplicación o si la BDD lo permite pedir el cambio de contraseña obligatorio después de la creación de un usuario**

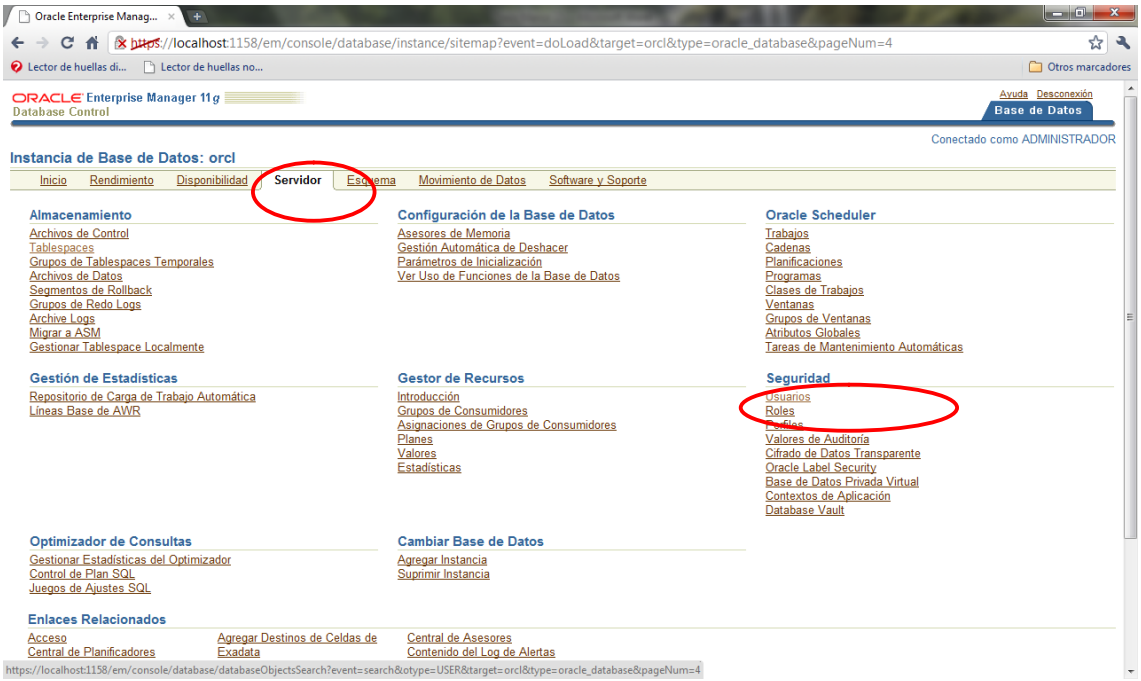
Implementación

Oracle otorga la posibilidad de administrar el cambio de contraseña después de la creación de un usuario, pero esto se lo puede implementar a través de una aplicación que funcione conjuntamente con este motor de base de datos. Eso depende de la estructura y de la forma en cómo se quiere administrar la seguridad de los sistemas informáticos.

En este caso lo vamos a realizar mediante la gestión del Motor de Base de Datos, y lo realizamos en la creación de un usuario.

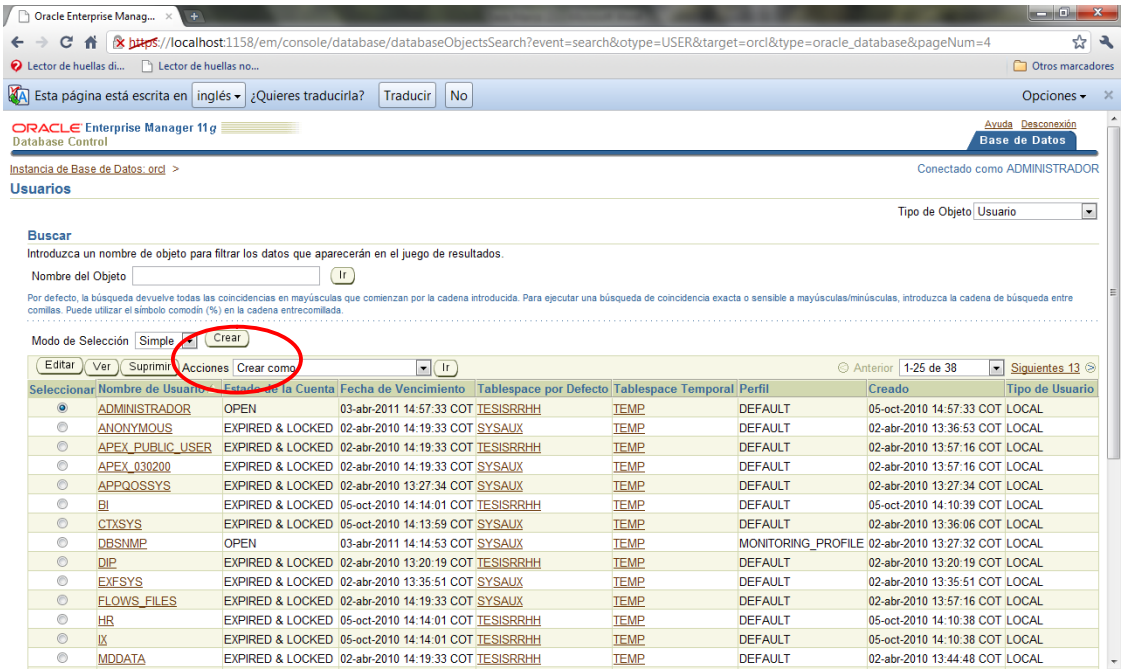
Para esto ingresamos a la interfaz de administración ORACLE ENTERPRISE MANAGER 11G (OEM) y seleccionamos la pestaña “SERVIDOR”.

Posteriormente seleccionamos en el grupo de SEGURIDAD el menú de USUARIOS.



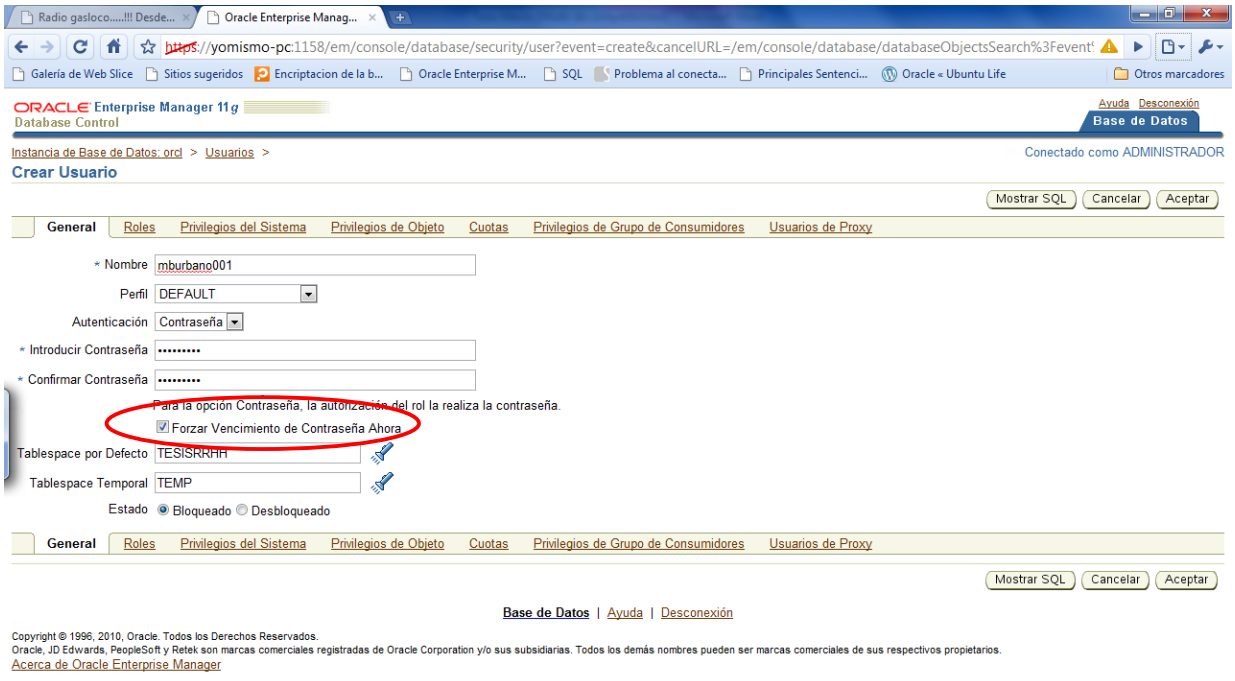
Realizado por Marco Burbano Fecha: 2010/08/23

En la nueva pantalla que se presenta podemos seleccionar la opción “Crear” para poder configurar la creación de un nuevo usuario.



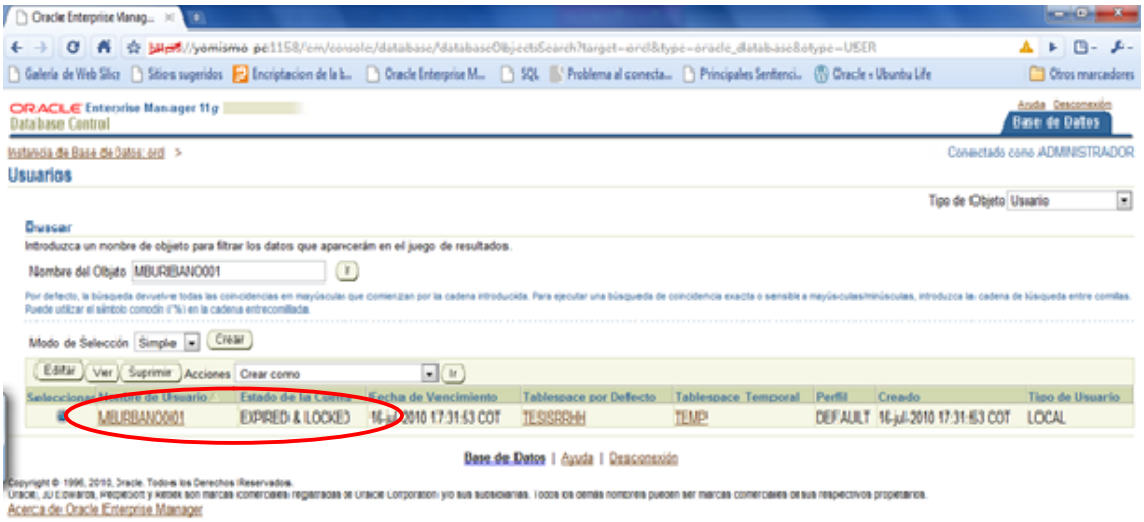
Realizado por Marco Burbano Fecha: 2010/08/23

De esta manera en la configuración de la nueva cuenta de usuario podemos forzar el vencimiento de la cuenta de usuario, para la siguiente vez que ingrese a la BDD se obligue el cambio de contraseña. Esto represente una buena forma de garantizar la seguridad de la BDD ya que se garantiza que solo la persona acreditada conoce la contraseña.



Realizado por Marco Burbano Fecha: 2010/08/23

Aquí podemos verificar que la cuenta se encuentra bloqueada y la contraseña expirada.

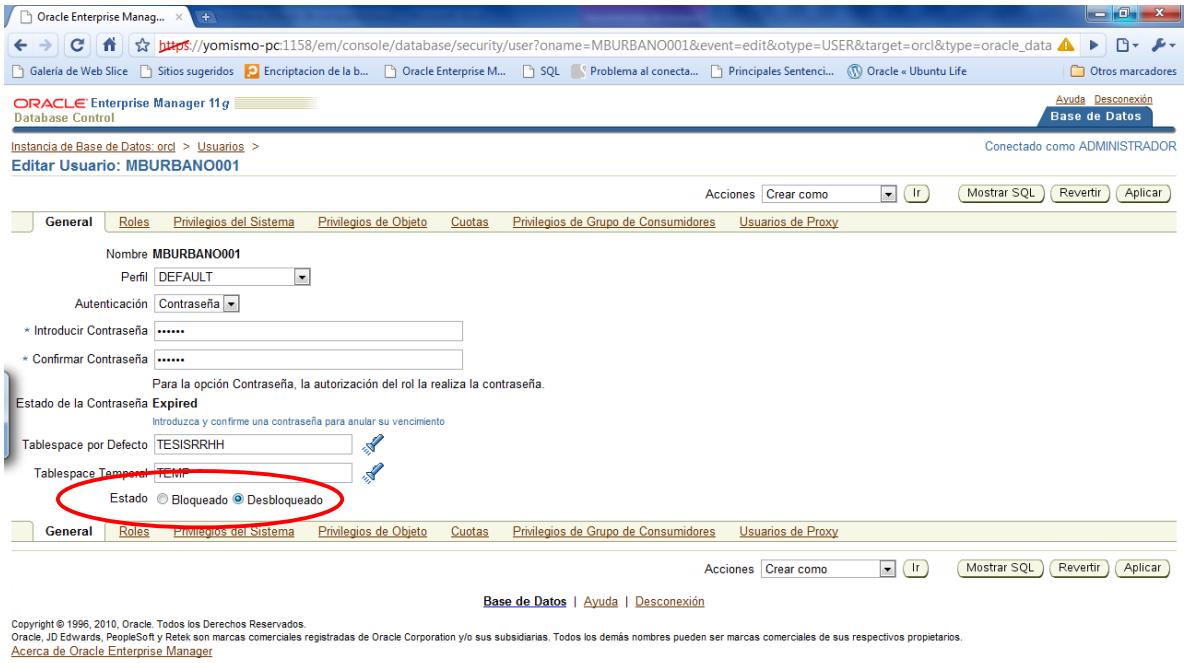


Realizado por Marco Burbano Fecha: 2010/08/23

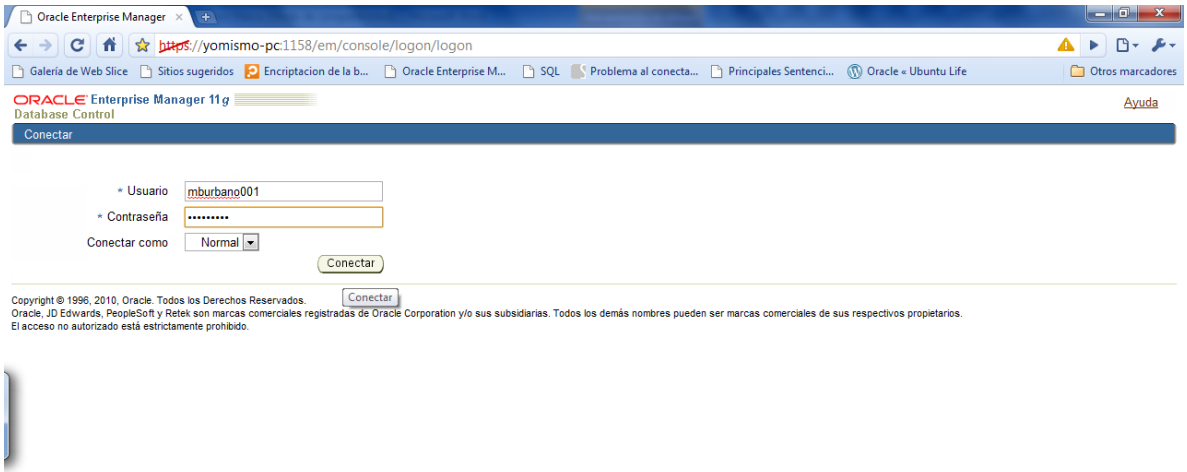
Podemos evidenciar claramente que mientras un usuario no sea desbloqueado, este no puede acceder a la BDD.

Realizado por Marco Burbano Fecha: 2010/08/23

Ahora nos registramos como un usuario con privilegios y procedemos a desbloquear el usuario.



Realizado por Marco Burbano Fecha: 2010/08/23



https://yomismo-pc:1158/em/console/logon/logon#

Realizado por Marco Burbano Fecha: 2010/08/23

4. Restringir el acceso de usuarios autorizados a la BDD

Implementación

Se debe definir un método de acceso a la BDD el cual puede ser mediante el uso de un usuario y contraseña autorizados o cualquier otro método eficaz.

En ORACLE tenemos la posibilidad de escoger varios métodos de autenticación a la Base de Datos, entre los cuales tenemos:

- Contraseña
- Externo
- Global

Contraseña. La propia base de datos mira que el usuario esté legitimado para acceder a la base de datos y comprueba que la contraseña sea correcta.

CREATE USER <usuario> IDENTIFIED BY <contraseña>;

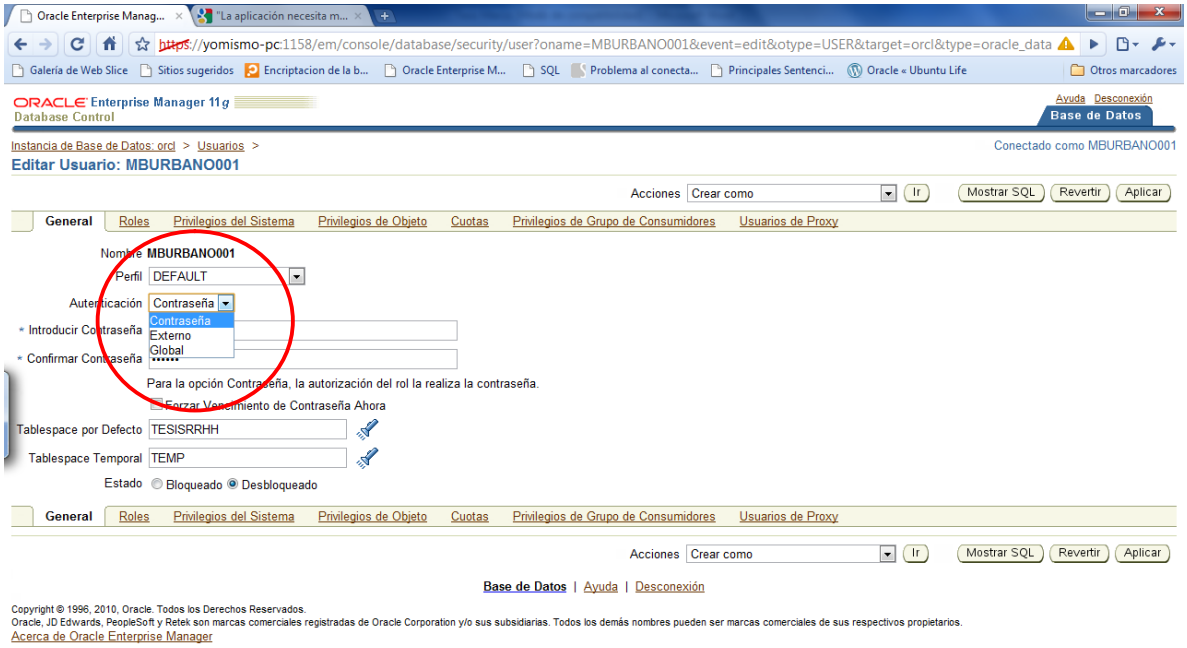
ALTER USER <usuario> IDENTIFIED BY <contraseña>;

Externo. Oracle sólo comprueba que el usuario esté legitimizado. La contraseña la comprueba el sistema operativo o la red. La cuenta debe empezar con el prefijo OPS$

CREATE USER ops$appl IDENTIFIED EXTERNALLY;

Global. Oracle sólo comprueba que el usuario esté legitimizado. La contraseña es validada por el producto Oracle Security Service.

CREATE USER scott IDENTIFIED GLOBALLY AS <directory>;



Realizado por Marco Burbano Fecha: 2010/08/23

5. **Para la ejecución de scripts o líneas de comando estas deben tener previa autorización más aún si estos tienen incluidas contraseñas de usuarios con poder en la BDD**

Implementación

Se debe definir políticas de aprobaciones para ejecutar scripts y con qué usuario y contraseña deben ser ejecutados además se debe definir quién es el responsable de la ejecución y revisión de estas rutinas.

Una buena práctica implementada por varias instituciones en el país ha sido la elaboración de bitácoras de ejecución de procesos batch ya sean scripts de BDD o de SO. En estas bitácoras lo más importante es definir quién, cuándo y de qué manera se ejecutó los scripts.

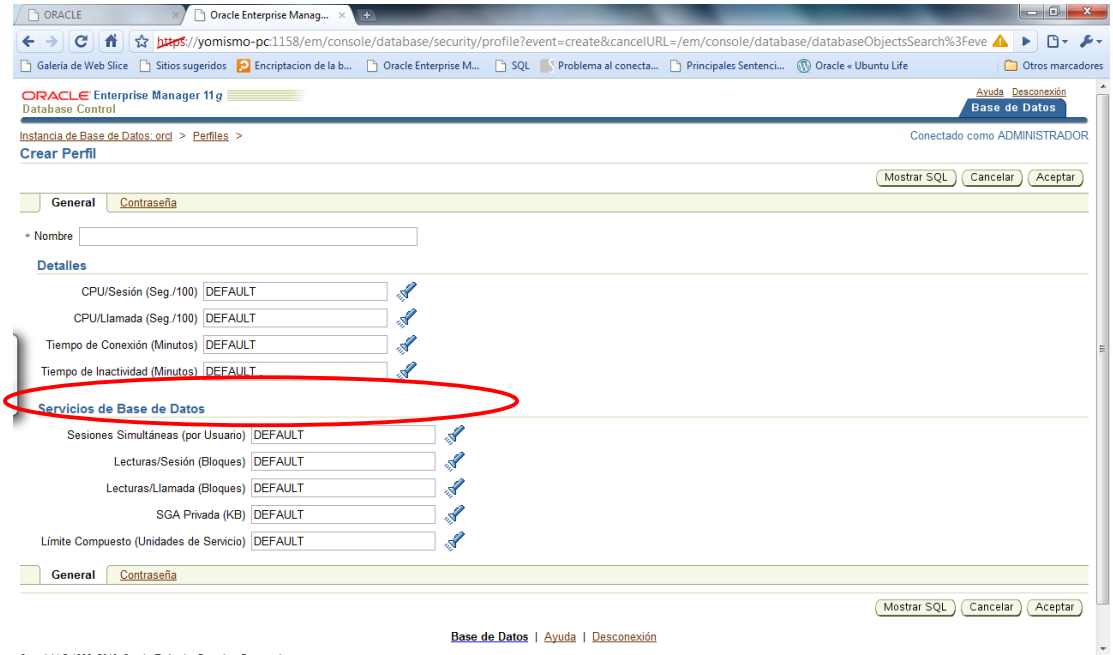
Para finalmente realizar la revisión de las ejecuciones de los scripts, ya que con esto se mantiene un fuerte control sobre los procesos de ejecución y los resultados obtenidos de lo realizado en el procedimiento.

6. Limitar el número de sesiones concurrentes por cada usuario de preferencia permitir una sola sesión por cada usuario

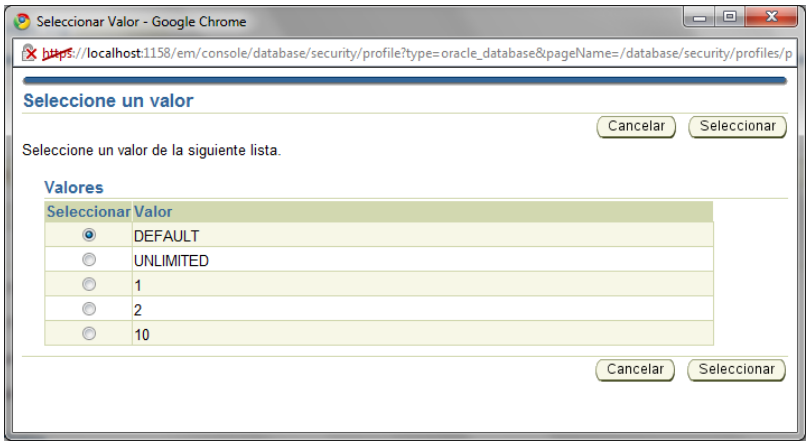
Implementación

Oracle permite esta restricción, al momento de crear un perfil para asignar a un usuario se nos permite configurar entre otras cosas el número de sesiones simultáneas que cada usuario es capaz de realizar.

Esta opción lo presenta en Enterprise Manager de Oracle en la Opción Perfiles en la administración de servidor.



Realizado por Marco Burbano Fecha: 2010/08/23



Realizado por Marco Burbano Fecha: 2010/08/23

7. La posibilidad de usar “GRANT OPTION” o “REVOKE” debe ser permitido solo a personal autorizado

Implementación

ORACLE permite esta restricción, esto se implementa en el momento de asignar a cada uno de los usuarios los respectivos privilegios de acuerdo a las funciones dentro de la organización.

Como se muestra esta opción solo se le otorga a los usuarios con la necesidad de generar y otorgar permisos en la BDD

En este caso como se trata de un usuario normal que tiene el cargo de asistente de RRHH no amerita mayores permisos.

Realizado por Marco Burbano Fecha: 2010/08/23

8. Denegar la posibilidad de conectarse remotamente a la BDD más aún si los usuarios tienen permisos para modificar la información sensitiva

Implementación[21]

Para acceder desde una base de datos Oracle a objetos de otra base de datos Oracle la manera más sencilla es utilizar un DBLINK (que sea la más sencilla no significa que siempre sea la más aconsejable, el abuso de los DBLINKS puede generar muchos problemas, tanto de rendimiento como de seguridad)

Para ello es necesario, con un usuario que posea el privilegio CREATE DATABASE LINK, crear el DBLINK en la base de datos origen (A) mediante una sencilla sentencia como la siguiente:

[21] Ver mas: http://www.stanford.edu/dept/itss/docs/oracle/10g/server.101/b10759/statements_5005.htm Obtenido el 28 de noviembre de 2010

Create database link LNK_DE_A_a_B connect to USUARIO identified by CONTRASEÑA USING 'B';

'LNK_DE_A_a_B' es el nombre del link, 'USUARIO' y 'CONTRASEÑA' son los identificadores del usuario que utilizará el link para conectarse, los permisos del cual heredarán todos los accesos a través del link, y B es el nombre de la instancia de la base de datos.

A través del DBLINK se puede conectar con los objetos de la base de datos remota con los permisos que tenga el usuario que se ha proporcionado en la sentencia de creación.

Para referenciar un objeto de la base de datos remota se ha de indicar el nombre del objeto, concatenado con el carácter '@' y el nombre que se le ha dado al DBLINK.

Ejemplo:

select * from TABLA@LNK_DE_A_a_B

9. **La información sensitiva almacenada en la Base de Datos debería ser encriptada. El no tener la información primordial encriptada aumenta el riesgo de accesos no autorizados.**

Implementación

Oracle guarda las contraseñas en la tabla interna **sys.user$**. En varias fuentes de información como Internet, foros o revistas se puede encontrar el típico error

que dice que Oracle guarda las contraseñas en la tabla DBA_USERS. En realidad DBA_USERS es una vista que apunta a la tabla sys.users$.

Oracle encripta las contraseñas de los usuarios del siguiente modo:
HASH en MD5 modificado del nombre del usuario (en mayúsculas) más la contraseña (en mayúsculas también).

Es decir, para el nombre de usuario "Fernando" y la contraseña "Alonso", Oracle hará lo siguiente:

1. Upper("Fernando") || Upper("AlonsO")) = FERNANDOALONSO
2. Aplica una función que obtiene el hash md5 pero con una modificación propia de Oracle, por lo que aunque apliquemos el HASH a esa cadena no nos dará lo que obtiene Oracle.

Hay que tener en cuenta que el mecanismo de encripción de la contraseña por parte de Oracle es unidireccional, esto quiere decir que Oracle almacena el HASH modificado pero nunca podrá obtener del HASH modificado la contraseña que el usuario introdujo. Es decir, no podrá desencriptarse la cadena encriptada.

Adicionalmente para encriptar información del negocio al que pertenece la BDD se pueden implementar algoritmos desarrollados In House o instalar paquetes de encripción comerciales como por ejemplo SGDB OBFUSCATION TOOLKIT[22]

[22]Ver más http://download-

10. Activar los registros de auditoría con la finalidad de llevar un control sobre las actividades realizadas en la BDD

Implementación

Oracle ofrece gran versatilidad en el momento de la configuración de los Logs de Auditoría, los cuales son de vital importancia para un eficiente monitoreo y ambiente de seguridad que debe existir en el manejo y almacenamiento de información.

El SGBD Oracle tiene la capacidad de auditar todas las acciones que tienen lugar en la BD. Se pueden auditar tres tipos de acciones:

- intentos de entrada en cuentas de la BD.
- accesos a los objetos de la BD.
- acciones sobre la BD.

La BD registra todos los intentos de acción, tanto los exitosos como los infructuosos, aunque es un parámetro configurable.

Para habilitar la capacidad de auditoría, se debe fijar el parámetro AUDIT_TRAIL en el fichero init.ora. Los registros de auditoría se almacenan en la tabla SYS.AUD$ o bien su gestión se deja al SO. Cuando se decide utilizar la tabla SYS.AUD$ esta debe revisarse periódicamente, por si

---

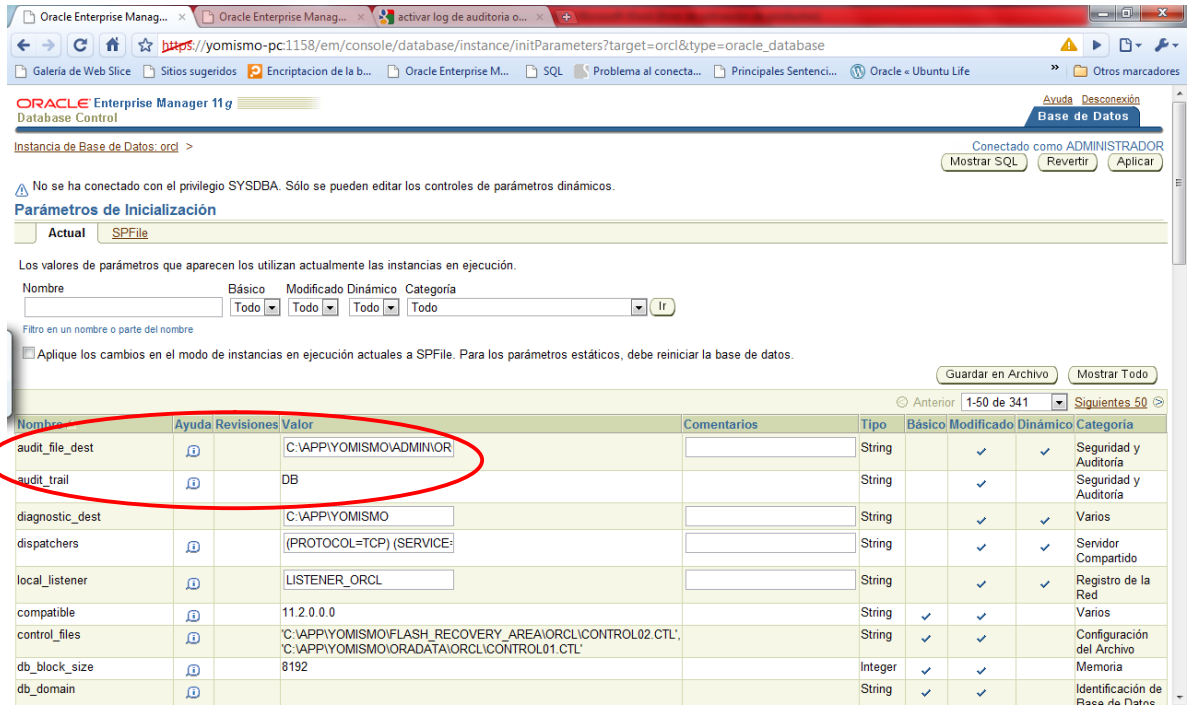
west.oracle.com/docs/cd/B14117_01/appdev.101/b10802/d_obtool.htm#1002213 obtenido el 13 de noviembre de 2010

hiciera falta truncarla debido a que su aumento de tamaño puede causar problemas de espacio en el *tablespace* SYSTEM. Los valores del parámetro AUDIT_TRAIL son los que se exponen en la Tabla 2.

Tabla 2[23]

Valor	*Descripción*
NONE	Deshabilita la auditoría
BD	Habilita la auditoría, escribiendo en la tabla SYS.AUD$.
OS	Habilita la auditoría, dejando al SO su gestión.

[23] Tabla 2: Parámetros de Auditoría obtenido de: http://www.infor.uva.es/~jvegas/cursos/bd/oraseg/oraseg.html#4.1 obtenido el 15 de noviembre de 2010



Realizado por Marco Burbano Fecha: 2010/08/23

10.1 Auditando Conexiones

Todo intento de conexión con la BD será registrado. El comando para iniciar la auditoría es

auditsession;

Para determinar si se deben registrar sólo los éxitos, o sólo los fracasos se pueden utilizar los siguientes comandos:

```
audit session whenever successful;
audit session whenever not successful;
```

Si los registros de auditoría se almacenan en la tabla SYS.AUD$, entonces pueden verse a través de la vista DBA_AUDIT_SESSION.

```
select
os_username,                /* nombre de usuario SO */
username,                   /* nombre de usuario BD */
terminal,
decode(returncode,'0','Conectado',
                '1005','Solo username, sin password',
                '1017','Password incorrecto',
returncode),  /* comprobacion de error */
to_char(timestamp,'DD-MON-YY HH24:MI:SS'), /* hora de entrada */
to_char(logoff_time,'DD-MON-YY HH24:MI:SS') /* hora de salida */
fromdba_audit_session;
```

Para deshabilitar la auditoria de las conexiones basta con ejecutar la siguiente sentencia:

```
noauditsession;
```

10.2 Auditando Acciones

Se puede auditar cualquier acción que afecte a cualquier objeto de la BD. Para facilitar la gestión, las acciones a auditar se encuentran agrupadas según los grupos que se muestran en la Tabla 3:

Tabla 3[24]

Grupo	*Comandos Auditados*
CLUSTER	Todas las sentencias que afecten a *clusters*.
DATABASE LINK	Todas las sentencias que afecten a enlaces de BD.
EXISTS	Todas las sentencias que fallen porque ya existe un objeto en la BD.
INDEX	Todas las sentencias que afecten a índices.
NOT EXISTS	Todas las sentencias que fallen porque un determinado objeto no existe.
PROCEDURE	Todas las sentencias que afecten a procedimientos.
PROFILE	Todas las sentencias que afecten a perfiles.
PUBLIC	Todas las sentencias que afecten a enlaces

[24] http://www.infor.uva.es/~jvegas/cursos/bd/oraseg/oraseg.html#4.1 Obtenido el 15 de noviembre de 2010

DATABASE LINK	públicos de BD.
PUBLIC SINONYM	Todas las sentencias que afecten a sinónimos públicos.
ROLE	Todas las sentencias que afecten a roles.
ROLLBACK SEGMENT	Todas las sentencias que afecten a segmentos de *rollback*.
SEQUENCE	Todas las sentencias que afecten a secuencias.
SESSION	Todas las sentencias de acceso a la BD.
SYNONYM	Todas las sentencias que afecten a sinónimos.
SYSTEM AUDIT	Todas las sentencias AUDIT y NOAUDIT.
SYSTEM GRANT	Todas las sentencias afecten a privilegios.
TABLE	Todas las sentencias que afecten a tablas.
TABLESPACE	Todas las sentencias que afecten a espacios de tablas.
TRIGGER	Todas las sentencias que afecten a disparadores.
USER	Todas las sentencias que afecten a las cuentas de usuarios.
VIEW	Todas las sentencias que afecten a vistas.

Por ejemplo, para auditar todas acciones que tienen que ver con las tablas sirve el siguiente comando:

audittable;

Y para deshabilitar la auditoría se utilizará el siguiente comando:

noaudittable;

También se puede afinar un poco más en la auditoría fijando un usuario concreto al que seguir la pista:

audittableby mburbano001;

Cada acción auditada recibe un código numérico al que se puede acceder a través de la vista AUDIT_ACTIONS. Una vez que conocemos el código de la acción, podemos utilizarlo para determinar cómo dicha acción ha afectado a un objeto, consultado la vista DBA_AUDIT_OBJECT.

10.3 Auditando Objetos

Además de la auditoría de acciones sobre los objetos, se puede seguir el rastro a las operaciones de manipulación de
tablas: SELECT, INSERT, UPDATE y DELETE. Estas auditorías se pueden hacer por sesión o por acceso.

Un ejemplo de sentencias de auditorías sobre objetos se puede ver en el siguiente grupo de sentencias:

audit insert on perez.emp;

audit all on perez.emp by session;

audit delete on perez.emp by access;

Los registros de auditoría se pueden ver en la misma

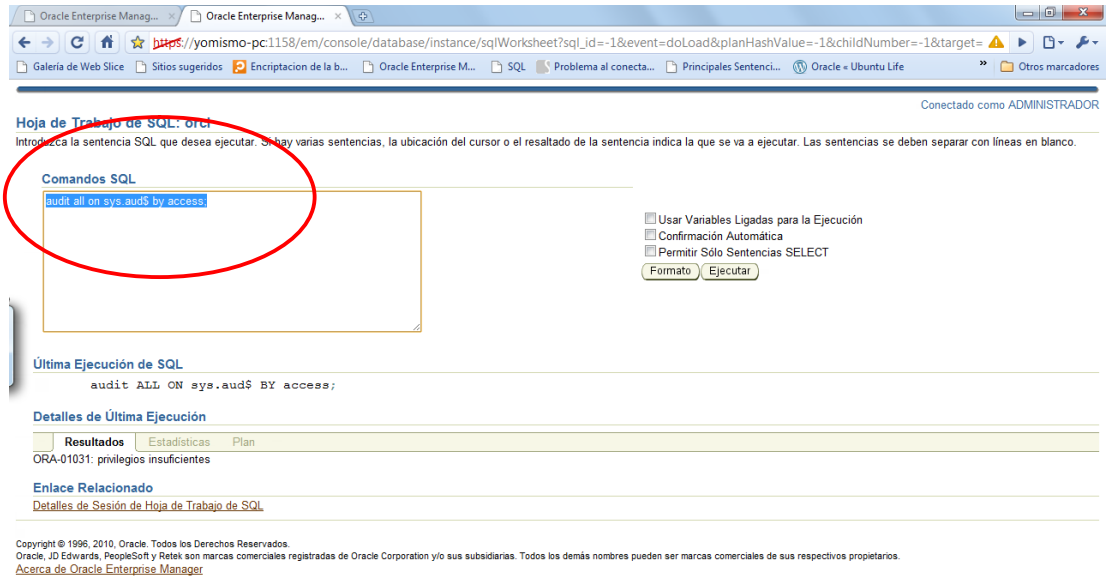
vista DBA_AUDIT_OBJECT anteriormente mencionada.

11. **El administrador de seguridad (no el DBA) debe supervisar la configuración de auditoría de base de datos. Sólo el administrador de seguridad debe tener acceso a los registros de auditoría.**

Implementación

Los registros de la tabla SYS.AUD$ pueden ser objeto de intentos de acceso para ser eliminados ya que pueden reflejar acciones no autorizadas en la BD. Así, resulta interesante reflejar ese tipo de acciones. Esto se consigue con el siguiente comando:

audit all on sys.aud$ by access;



Realizado por Marco Burbano Fecha: 2010/08/23

De este modo cualquier acción contra la tabla SYS.AUD$ quedará registrado. Además, las acciones contra la tabla SYS.AUD$ sólo pueden ser borradas por los usuarios que puedan conectarse como INTERNAL.

12. Asegurar que los controles de Respaldo y Restauración de la Base de Datos garantiza la disponibilidad de los datos los cuales se pueden recuperar por completo

Implementación[25]

A pesar de algunas recomendaciones preventivas y de recuperación mencionadas en este informe, cada organización de IT debe implementar un

[25] Obtenido de: http://www.oracle.com/technology/global/lad-es/documentation/collaterals/Oracle%20Database%2011g%20High%20Availability_cast_.pdf Consultado el 5 de noviembre de 2010

procedimiento para los backups de datos. Existen casos donde se producen múltiples fallas al mismo tiempo, aunque es poco común, y el administrador debe poder recuperar los datos críticos de negocio desde el backup. Oracle ofrece herramientas estándar del sector para hacer un backup eficiente y adecuado de los datos, restaurar los datos de backups anteriores, y recuperar los datos hasta el momento anterior a la falla.

12.1 Recovery Manager (RMAN)

Las bases de datos grandes pueden estar compuestas por cientos de archivos diseminados en muchos puntos de soporte, haciendo que las actividades de backup sean extremadamente desafiantes. Descuidar o pasar por alto incluso un solo archivo crítico de un backup puede hacer que todo el backup de base de datos sea inútil. Debido a que es muy frecuente el caso, los backups incompletos no se detectan hasta que se necesitan en una emergencia. Oracle Recovery Manager (RMAN) es la herramienta compuesta que administra el backup, la restauración y los procesos de recuperación de la base de datos. RMAN mantiene las políticas configurables de backup y recuperación y guarda registros históricos de todas las actividades de backup y recuperación de la base de datos. A través de sus características integrales, RMAN garantiza que todos los archivos requeridos para restaurar y recuperar exitosamente una base de datos sean incluidos en backups completos de la base de datos. Asimismo, mediante las operaciones RMAN de backup, todos los bloques de datos son analizados para garantizar que los bloques corruptos no se propaguen en los archivos de backup.

Las mejoras en RMAN han hecho que el backup de grandes bases de datos sea un proceso eficiente y directo. RMAN aprovecha las capacidades Block Tracking para aumentar el desempeño de los backups incrementales. Hacer solamente un backup de los bloques que han cambiado a partir del último backup reduce enormemente el tiempo y los gastos generales del backup RMAN. En Oracle Database 11g, las capacidades Block Tracking ahora están activadas en bases de datos standby administradas. Debido a que el tamaño de las bases de datos empresariales sigue creciendo– se ha vuelto más ventajoso aprovechar BigfileTablespaces. Un Espacio de tabla Bigfile está conformado por un solo archivo grande en lugar de varios archivos más pequeños, permitiendo que las bases de datos Oracle escalen hasta 8 exabytes de tamaño. Para aumentar el desempeño de las operaciones de backup y recuperación de BigfileTablespaces – RMAN en Oracle Database 11g puede realizar operaciones paralelas dentro del archivo para el backup y la recuperación.

Muchas empresas crean clones o copias de sus bases de datos de producción para utilizarlas en pruebas, garantía de calidad y para generar una base de datos standby. RMAN ha tenido durante mucho tiempo la capacidad de clonar una base de datos utilizando backups RMAN existentes mediante la funcionalidad DUPLICATE DATABASE. Antes de Oracle Database 11g, se debía acceder a los archivos de backup en el host de la base de datos clonada. La duplicación de Oracle Database 11g basada en redes duplicará la base de datos de origen en la base de datos clon sin la necesidad de que la base de datos de origen tenga backups existentes. En cambio, la duplicación basada en redes clonará transparentemente los archivos necesarios directamente desde el origen

hasta el clon. Oracle Database 11g tiene una estrecha integración con Microsoft Virtual Shadow CopyService (VSS). Concretamente, Microsoft Virtual Shadow CopyService es una estructura de tecnología que permite que las aplicaciones continúen escribiendo a volúmenes de disco mientras se ejecutan los backups consistentes de esos volúmenes. Oracle VSS Writer, un programa ejecutable separado que funciona en sistemas Windows, actuará como coordinador entre la base de datos Oracle y otros componentes VSS. Por ejemplo, Oracle VSS Writer colocará los archivos de base de datos en modo backup activo para permitir que los componentes VSS hagan una copia recuperable del archivo de datos en una snapshot VSS. Oracle VSS Writer aprovechará RMAN como la herramienta utilizada para realizar recuperaciones en los archivos almacenados desde una snapshot VSS. Además, RMAN ha sido mejorado para utilizar snapshots VSS como origen para backups almacenados en el Área de Recuperación Flash.

12.2 Data RecoveryAdvisor

Cuando surge una situación inconcebible y los datos de negocio críticos están en peligro, todas las opciones de recuperación y reparación necesitan evaluarse para garantizar una recuperación segura y rápida. Estas situaciones pueden ser estresantes y a menudo pueden producirse en medio de la noche. Las investigaciones muestran que los administradores invierten la mayoría del *Tiempo de Reparación* realizando investigaciones sobre qué, por qué y cómo se han comprometido los datos. Los administradores necesitan buscar a través de volúmenes de información para identificar los errores relevantes, las alertas y rastrear los archivos.

Oracle Database 11g Data RecoveryAdvisor, creado para minimizar el tiempo invertido en las fases de investigación y planificación de recuperación, reduce la incertidumbre y confusión durante un corte de servicio. Fuertemente integrado con otras características de alta disponibilidad de Oracle, como Data Guard y RMAN,

Data RecoveryAdvisor analiza todos los escenarios de recuperación con rapidez y precisión. Mediante esta integración, el asesor puede identificar qué opciones de recuperación son probables dadas las condiciones específicas. Las posibles opciones de recuperación son presentadas al administrador, clasificadas sobre la base del tiempo de recuperación y la pérdida de datos. Data RecoveryAdvisor puede configurarse para implementar automáticamente las mejores opciones de recuperación, reduciendo así cualquier dependencia sobre el administrador.

Muchos escenarios de desastre se pueden reducir con el análisis preciso de errores y los archivos de rastreo que se presentan antes de un corte de servicio. Por consiguiente, el Asesor de Recuperación de Datos analiza automática y continuamente la condición de la base de datos a través de varios controles de estado. A medida que el asesor identifica los síntomas que podrían ser precursores del corte de servicio de una base de datos, el administrador puede aceptar el consejo de recuperación y tomar las medidas necesarias para solucionar el problema relacionado y evitar el tiempo de baja del sistema.

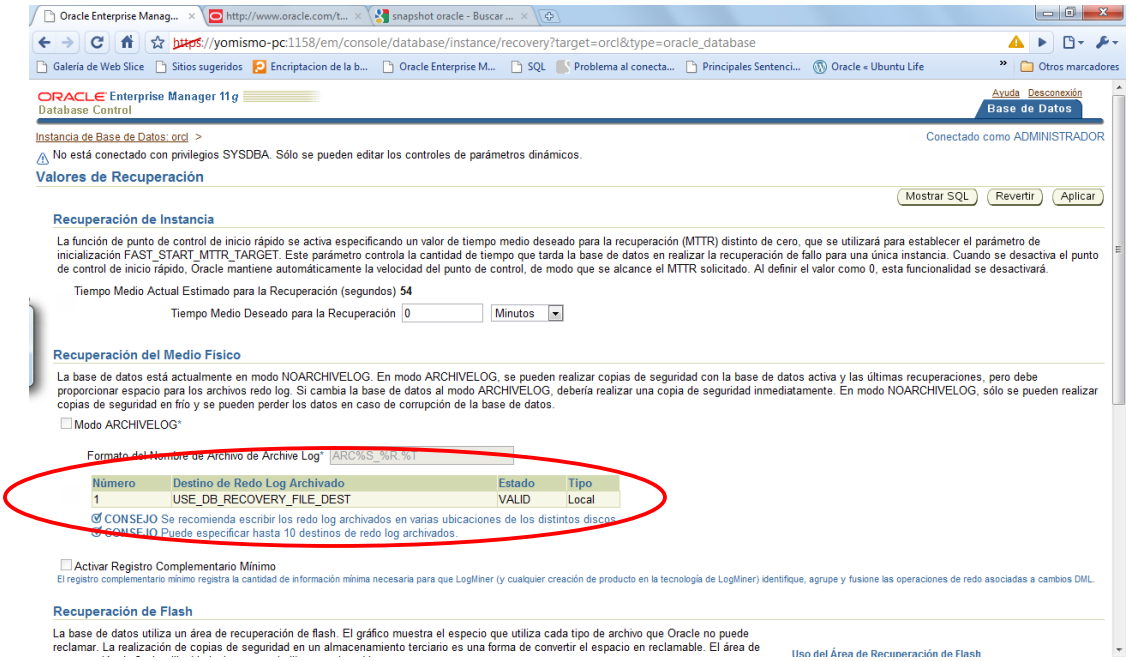
12.3 Oracle SecureBackup

Oracle SecureBackup – una nueva oferta de productos Oracle – ofrece administración centralizada de backup en cinta para todos los entornos Oracle,

con inclusión de las bases de datos y los sistemas de archivo. Oracle SecureBackupofrece a los clientes una solución de backup en cinta altamente segura, económica y de alto desempeño. Gracias a su estrecha integración con Oracle Database, Oracle SecureBackup puede realizar un backup de Oracle Database hasta un 25% más rápido que la competencia líder. Esto se logra al realizar llamadas directas en el motor de la base de datos y a través de algoritmos eficientes que omiten los bloques de datos no utilizados. Esta ventaja de desempeño solo continuará ampliándose en el futuro en la medida en que Oracle SecureBackup se integre progresivamente con el motor de base de datos, generando así optimizaciones especiales para mejorar aún más el desempeño del backup.

Oracle SecureBackup también se integra con Oracle Enterprise Manager – nuestra herramienta administrativa GUI basada en la Web – brindando a los administradores una incomparable facilidad de uso respecto de la configuración de los backups en cinta o la restauración/recuperación de los datos de cintas.

Para realizar una correcta administración del procedimiento de backup y restauración de la BDD es necesario tener los privilegios correctamente asignados para llevar a cabo la configuración de este procedimiento.



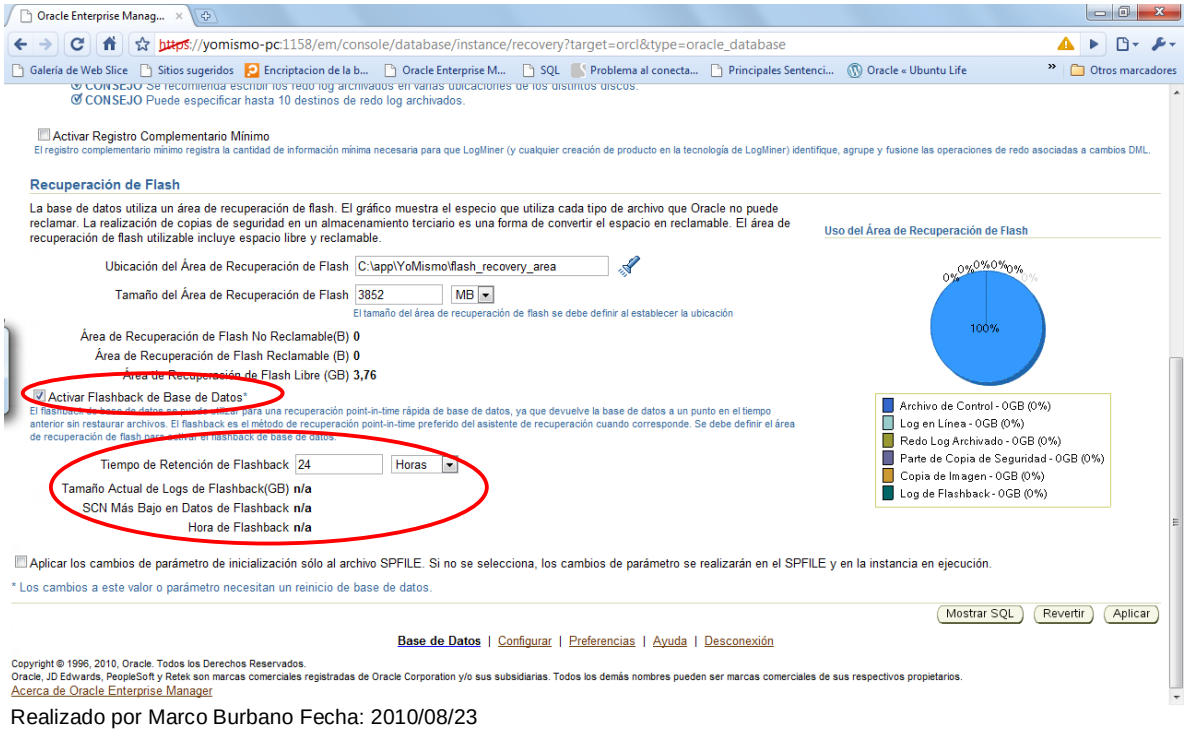
Realizado por Marco Burbano Fecha: 2010/08/23

Posteriormente será necesario configurar varios parámetros para de esta manera aumentar la eficiencia del procedimiento de Backup y posteriormente la restauración de los Backups obtenidos.

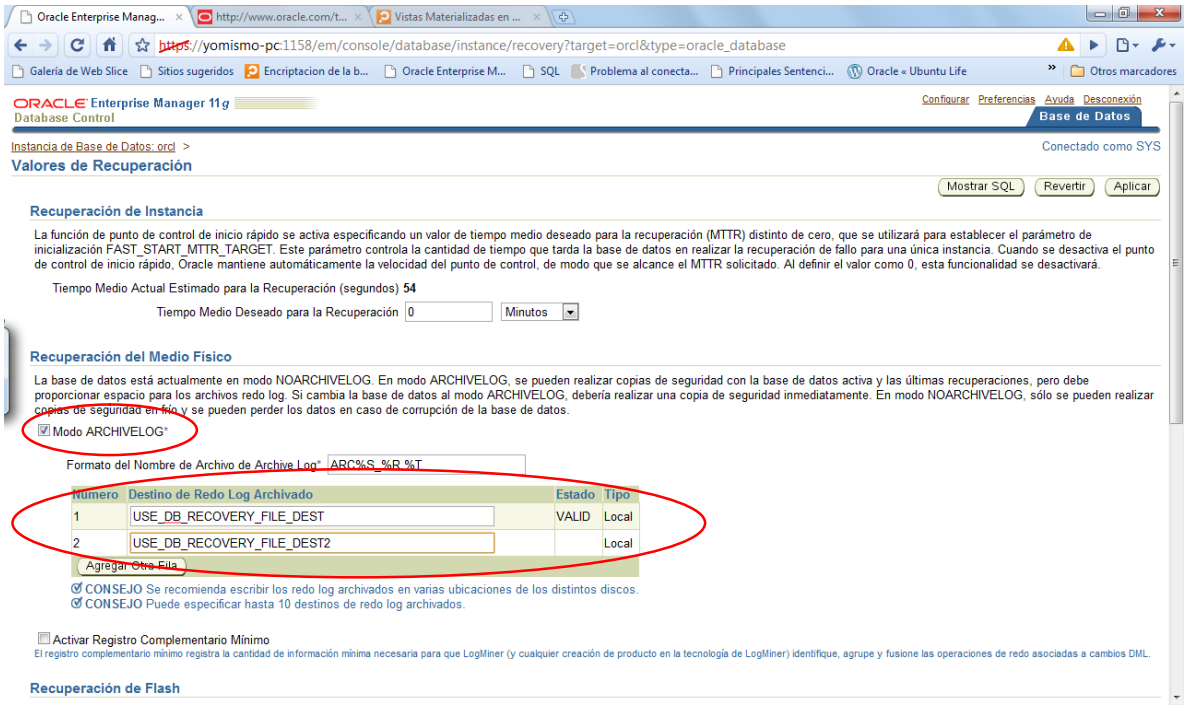
Uno de los principales parámetros que son necesario habilitar son la configuración del MODO ARCHIVELOG en el cual se pueden realizar copias de seguridad con la base de datos activa y las últimas recuperaciones, pero debe proporcionar espacio para los archivos redo log. Si cambia la base de datos al modo ARCHIVELOG, debería realizar una copia de seguridad inmediatamente. En modo NOARCHIVELOG, sólo se pueden realizar copias de seguridad en frío y se pueden perder los datos en caso de corrupción de la base de datos.

Adicionalmente como una buena práctica sugerida por Oracle es escribir los Redo Logs en varias localidades en distintos lugares físicos.

Adicionalmente como una buena medida para garantizar la completa recuperación de la base de datos así como también de la integridad de la información almacenada en la Base de Datos. Es la implementación del Flashback de Base de Datos lo cual se puede utilizar para una recuperación point-in-time rápida de base de datos, ya que devuelve la base de datos a un punto en el tiempo anterior sin restaurar archivos. El flashback es el método de recuperación point-in-time preferido del asistente de recuperación cuando corresponde. Se debe definir el área de recuperación de flash para activar el flashback de base de datos.



Realizado por Marco Burbano Fecha: 2010/08/23



Realizado por Marco Burbano Fecha: 2010/08/23

13. Realizar una correcta asignación de privilegios a cada uno de los usuarios

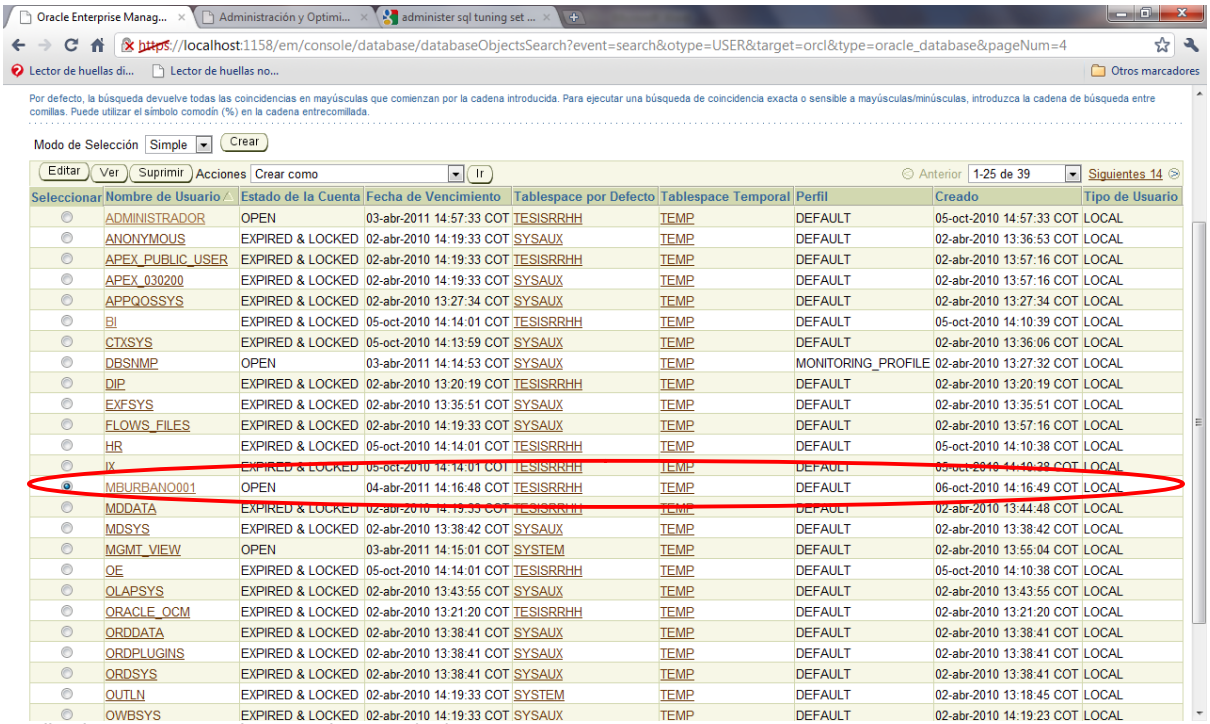
Implementación

Implementar procedimientos rigurosos para la asignación de perfiles y roles en la BDD, cada organización debe definir procedimientos y políticas que debe seguirse con la finalidad de minimizar el riesgo de asignar permisos incorrectos y que esto impacte directamente la información que se almacena en el motor de BDD.

En este caso crearemos un asistente de RRHH quien solo tiene el privilegio de conectarse a la BDD y visualizar la información de las tabas, mas no editar o eliminar la información almacenada.

Para esto Ingresamos a la Sección de Usuarios y consultamos el usuario MBURBANO001

El cuál es nuestro asistente de RRHH



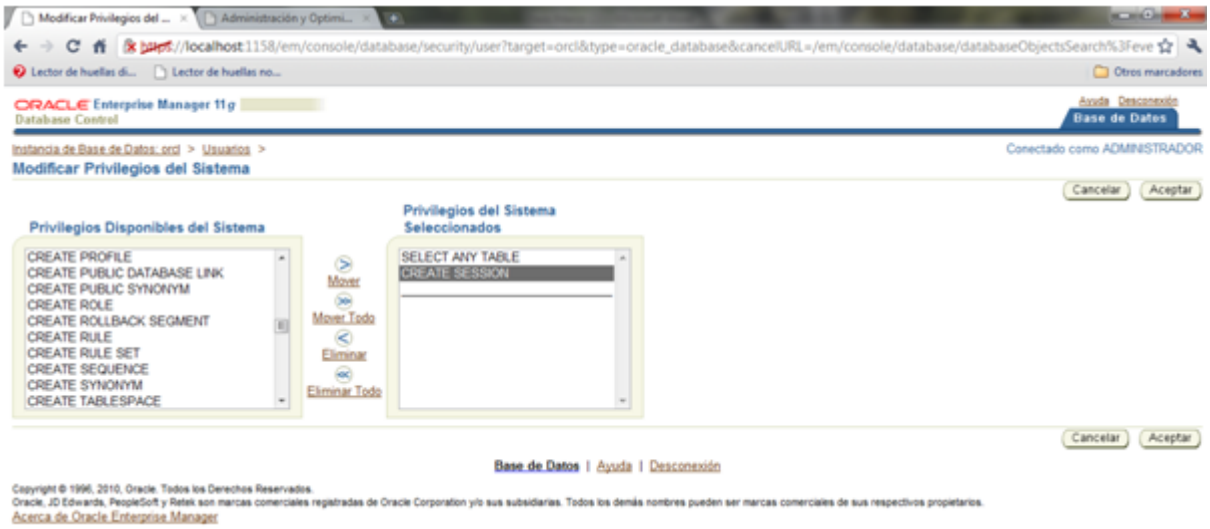
Seleccionar	Nombre de Usuario	Estado de la Cuenta	Fecha de Vencimiento	Tablespace por Defecto	Tablespace Temporal	Perfil	Creado	Tipo de Usuario
	ADMINISTRADOR	OPEN	03-abr-2011 14:57:33 COT	TESISRRHH	TEMP	DEFAULT	05-oct-2010 14:57:33 COT	LOCAL
	ANONYMOUS	EXPIRED & LOCKED	02-abr-2010 14:19:33 COT	SYSAUX	TEMP	DEFAULT	02-abr-2010 13:36:53 COT	LOCAL
	APEX_PUBLIC_USER	EXPIRED & LOCKED	02-abr-2010 14:19:33 COT	TESISRRHH	TEMP	DEFAULT	02-abr-2010 13:57:16 COT	LOCAL
	APEX_030200	EXPIRED & LOCKED	02-abr-2010 14:19:33 COT	SYSAUX	TEMP	DEFAULT	02-abr-2010 13:57:16 COT	LOCAL
	APPQOSSYS	EXPIRED & LOCKED	02-abr-2010 13:27:34 COT	SYSAUX	TEMP	DEFAULT	02-abr-2010 13:27:34 COT	LOCAL
	BI	EXPIRED & LOCKED	05-oct-2010 14:14:01 COT	TESISRRHH	TEMP	DEFAULT	05-oct-2010 14:10:39 COT	LOCAL
	CTXSYS	EXPIRED & LOCKED	05-oct-2010 14:13:59 COT	SYSAUX	TEMP	DEFAULT	02-abr-2010 13:36:06 COT	LOCAL
	DBSNMP	OPEN	03-abr-2011 14:14:53 COT	SYSAUX	TEMP	MONITORING_PROFILE	02-abr-2010 13:27:32 COT	LOCAL
	DIP	EXPIRED & LOCKED	02-abr-2010 13:20:19 COT	TESISRRHH	TEMP	DEFAULT	02-abr-2010 13:20:19 COT	LOCAL
	EXFSYS	EXPIRED & LOCKED	02-abr-2010 13:35:51 COT	SYSAUX	TEMP	DEFAULT	02-abr-2010 13:35:51 COT	LOCAL
	FLOWS_FILES	EXPIRED & LOCKED	02-abr-2010 14:19:33 COT	SYSAUX	TEMP	DEFAULT	02-abr-2010 13:57:16 COT	LOCAL
	HR	EXPIRED & LOCKED	05-oct-2010 14:14:01 COT	TESISRRHH	TEMP	DEFAULT	05-oct-2010 14:10:38 COT	LOCAL
	IX	EXPIRED & LOCKED	05-oct-2010 14:14:01 COT	TESISRRHH	TEMP	DEFAULT	05-oct-2010 14:10:38 COT	LOCAL
	MBURBANO001	OPEN	04-abr-2011 14:16:48 COT	TESISRRHH	TEMP	DEFAULT	06-oct-2010 14:16:49 COT	LOCAL
	MDDATA	EXPIRED & LOCKED	02-abr-2010 14:19:33 COT	TESISRRHH	TEMP	DEFAULT	02-abr-2010 13:44:48 COT	LOCAL
	MDSYS	EXPIRED & LOCKED	02-abr-2010 13:38:42 COT	SYSAUX	TEMP	DEFAULT	02-abr-2010 13:38:42 COT	LOCAL
	MGMT_VIEW	OPEN	03-abr-2011 14:15:01 COT	SYSTEM	TEMP	DEFAULT	02-abr-2010 13:55:04 COT	LOCAL
	OE	EXPIRED & LOCKED	05-oct-2010 14:14:01 COT	TESISRRHH	TEMP	DEFAULT	05-oct-2010 14:10:38 COT	LOCAL
	OLAPSYS	EXPIRED & LOCKED	02-abr-2010 13:43:55 COT	SYSAUX	TEMP	DEFAULT	02-abr-2010 13:43:55 COT	LOCAL
	ORACLE_OCM	EXPIRED & LOCKED	02-abr-2010 13:21:20 COT	TESISRRHH	TEMP	DEFAULT	02-abr-2010 13:21:20 COT	LOCAL
	ORDDATA	EXPIRED & LOCKED	02-abr-2010 13:38:41 COT	SYSAUX	TEMP	DEFAULT	02-abr-2010 13:38:41 COT	LOCAL
	ORDPLUGINS	EXPIRED & LOCKED	02-abr-2010 13:38:41 COT	SYSAUX	TEMP	DEFAULT	02-abr-2010 13:38:41 COT	LOCAL
	ORDSYS	EXPIRED & LOCKED	02-abr-2010 13:38:41 COT	SYSAUX	TEMP	DEFAULT	02-abr-2010 13:38:41 COT	LOCAL
	OUTLN	EXPIRED & LOCKED	02-abr-2010 14:19:33 COT	SYSTEM	TEMP	DEFAULT	02-abr-2010 13:18:45 COT	LOCAL
	OWBSYS	EXPIRED & LOCKED	02-abr-2010 14:19:33 COT	SYSAUX	TEMP	DEFAULT	02-abr-2010 14:19:23 COT	LOCAL

Realizado por Marco Burbano Fecha: 2010/09/13

Posteriormente procedemos a seleccionar la opción de EDITAR y nos dirigimos a la pestana de PRIVILEGIOS DEL SISTEMA y Seleccionamos la opción Editar Lista.

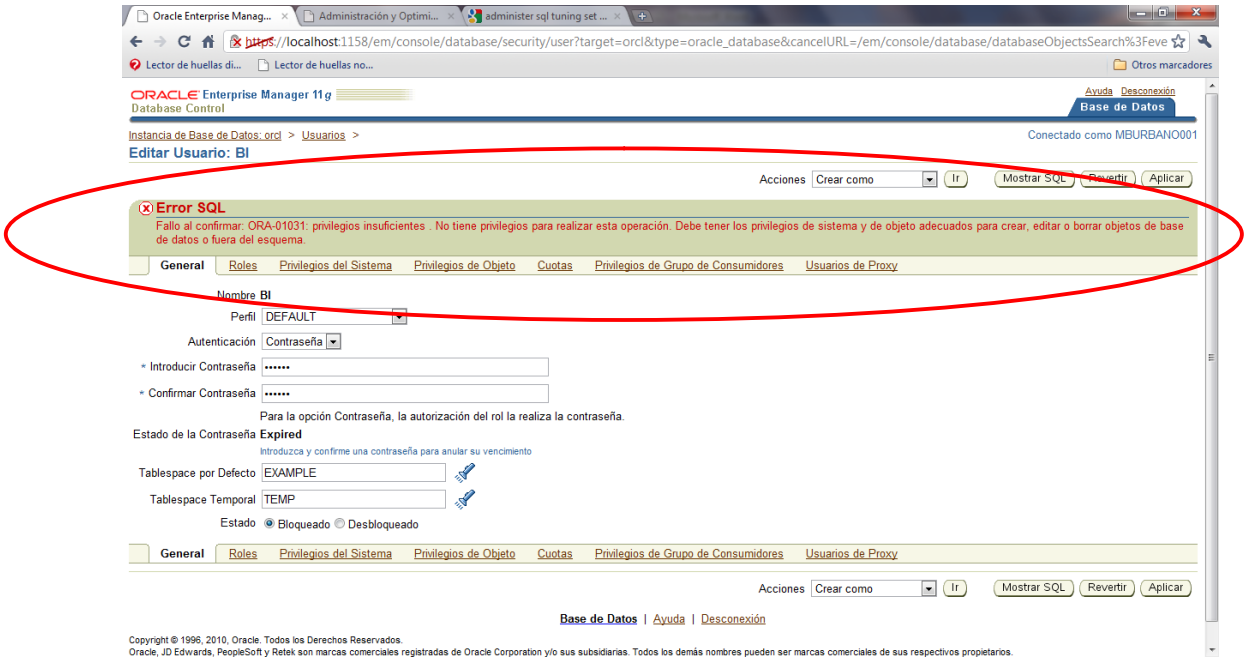
Realizado por Marco Burbano Fecha: 2010/09/13

Seguidamente seleccionamos los privilegios adecuados para este usuario que para este usuario solo necesitamos agregar los privilegios de visualización y conexión a la BDD.



Realizado por Marco Burbano Fecha: 2010/09/13

Ahora revisamos que estos privilegios solo permiten la visualización y conexión a la BDD, con lo cual no podemos realzar o modificar ningún otro parámetro.



Realizado por Marco Burbano Fecha: 2010/09/13

14. \Todos los identificadores de usuario debe ser único e identificar un único usuario y debe respetar la norma de denominación corporativa.

Implementación

Definir políticas para el acceso a la BDD, eso lo debe establecer el gerente de TI con una delegación entre los cuales debe estar el Gerente General con la finalidad de otorgarle importancia a las políticas que se establecen.

Es importante que cada vez que un empleado ingrese a la institución se le den a conocer las políticas de la seguridad de información desarrolladas en el área de IT.

Entre las cuales se debe mencionar que queda prohibido compartir la clave de acceso a los sistemas o BDD.

15. Restringir el uso de líneas de comando a usuarios no autorizados

Implementación

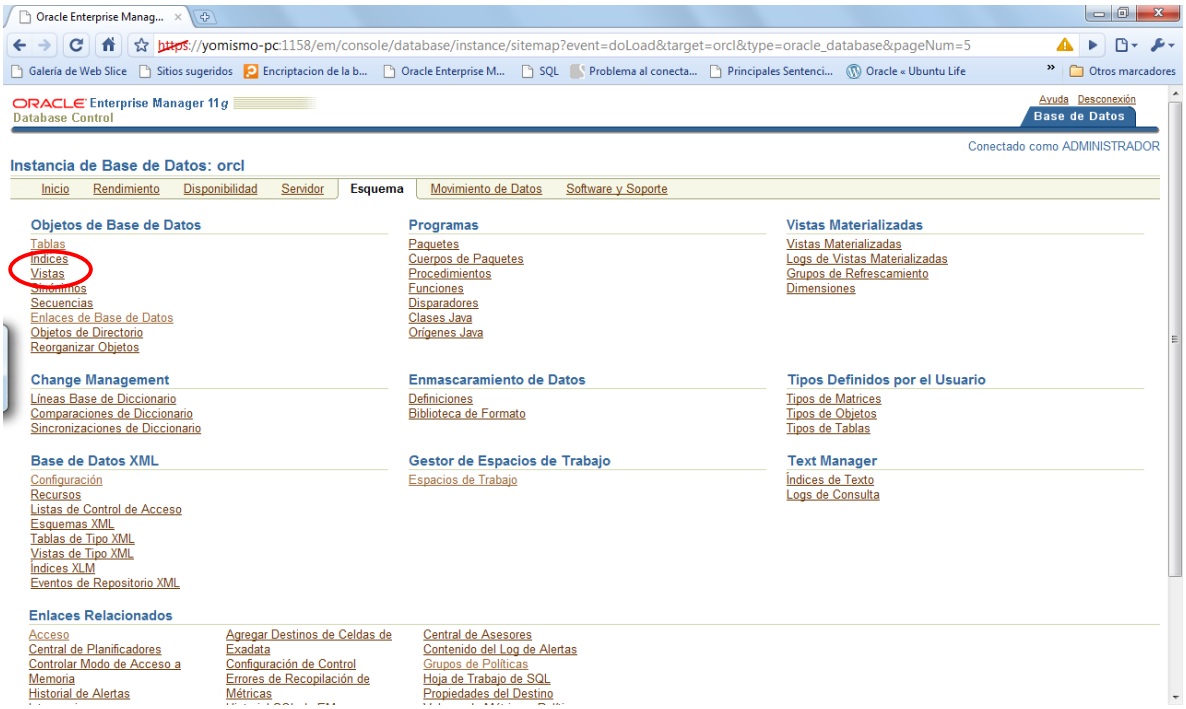
El acceso a líneas de comando o ejecución de querys sin ningún tipo de control disminuye la seguridad e incluso la disponibilidad de la Base de Datos, es por esto que es recomendable limitar la instalación de interfaces que puedan conectarse con la BDD por ejemplo iSQLPlus, TOAD, Oracle Developer o cualquier herramienta que permita modificar cualquier parámetro o información de la base de datos.

16. Aplicación de vistas con la finalidad de restringir la información a usuarios finales

Implementación

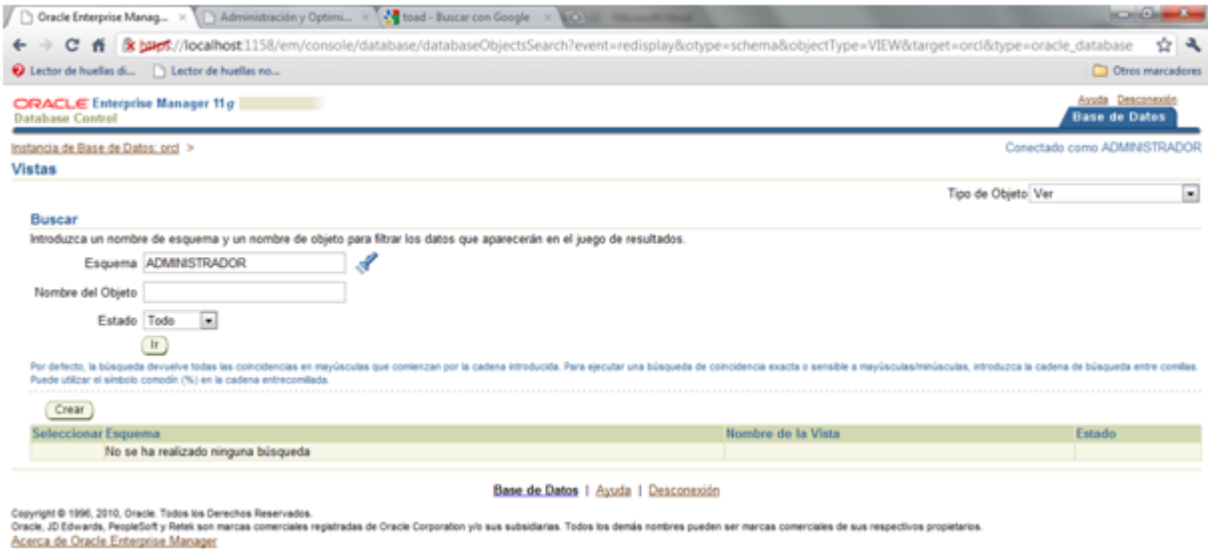
Oracle permite la realización de vistas. Lo cual nos ayuda a administrar de mejor manera la información, así como también presentar al usuario solo la información necesaria.

Para crear una vista mediante la consola de administración OEM nos dirigimos a la pestaña Esquema y en la sección Objetos de Base de Datos seleccionamos el Menú Vistas.



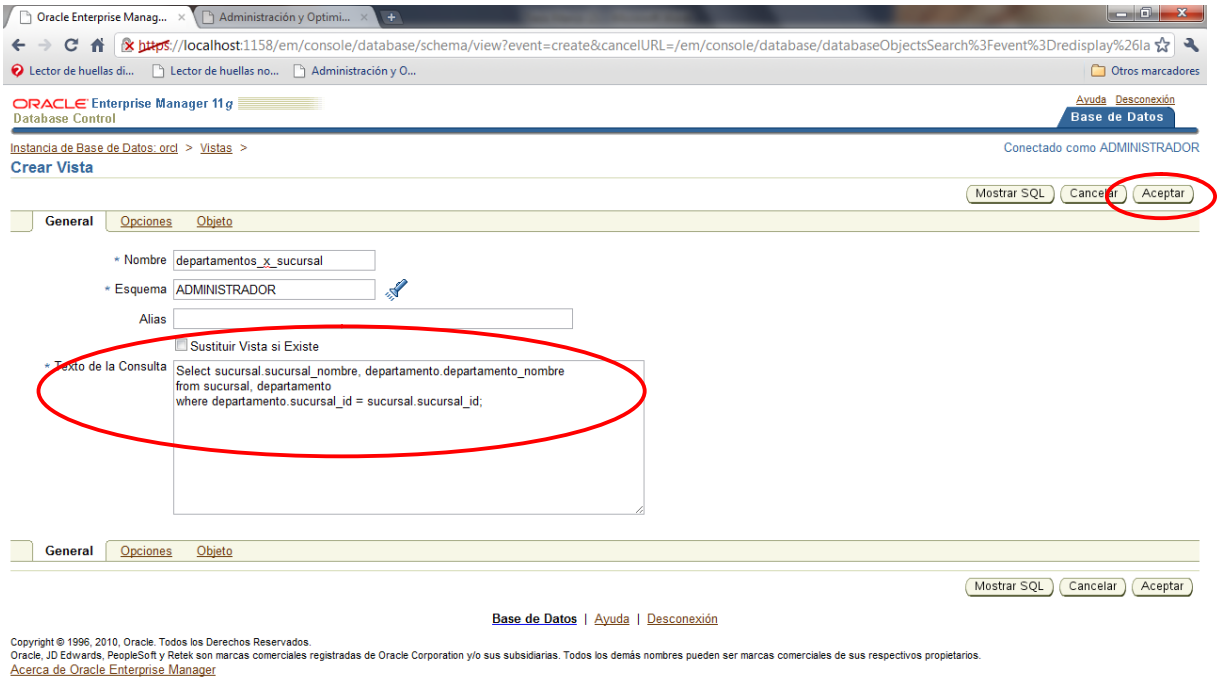
Realizado por Marco Burbano Fecha: 2010/09/30

Aparecerá una nueva pantalla donde presionamos el botón “Crear”.



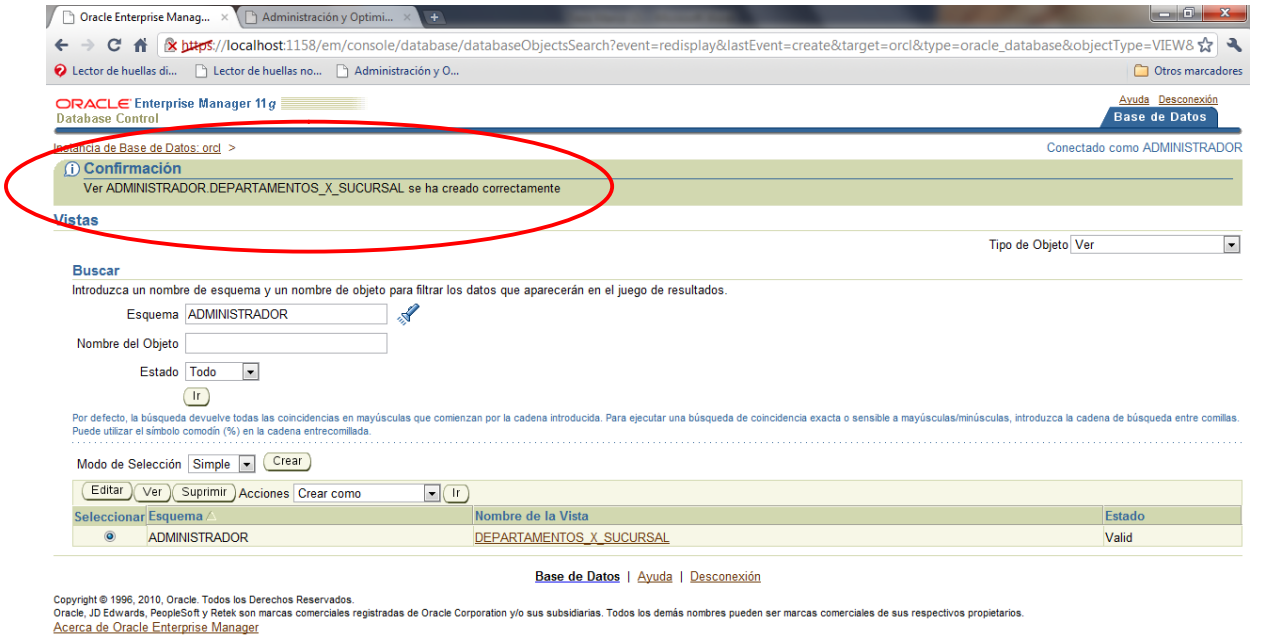
Realizado por Marco Burbano Fecha: 2010/09/30

Llenamos la información como el nombre y la consulta Select con la cual realizaremos la Vista y presionamos el botón “Aceptar”.



Realizado por Marco Burbano Fecha: 2010/09/30

Si la consulta y la información proporcionada es correcta obtendremos un mensaje de confirmación de que la vista ha sido creada correctamente.



Realizado por Marco Burbano Fecha: 2010/09/30

No olvidar que también es posible crear y configurar los parámetros u objetos de la BDD mediante consultas o comandos es por esto que crearemos la vista mediante Consultas. Para esto primero definimos la consulta que se va a transformar en una vista esto se lo puede realizar mediante SQL.

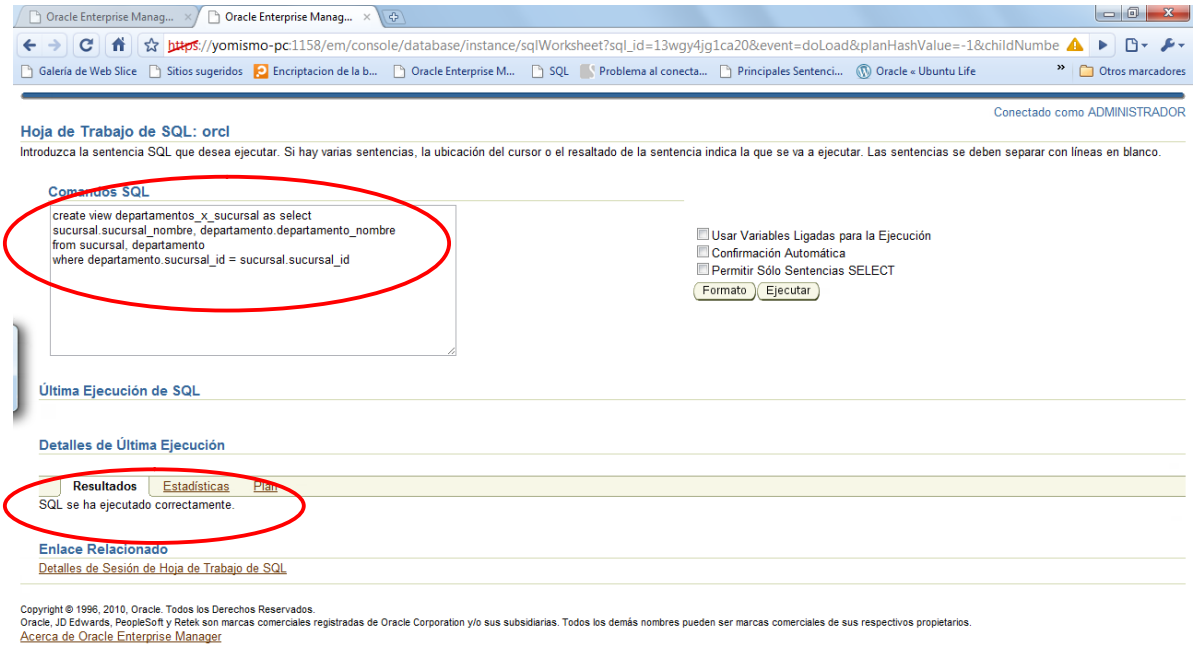
La vista será la siguiente:

Create view departamentos_x_sucursal as

Select sucursal.sucursal_nombre, departamento.departamento_nombre

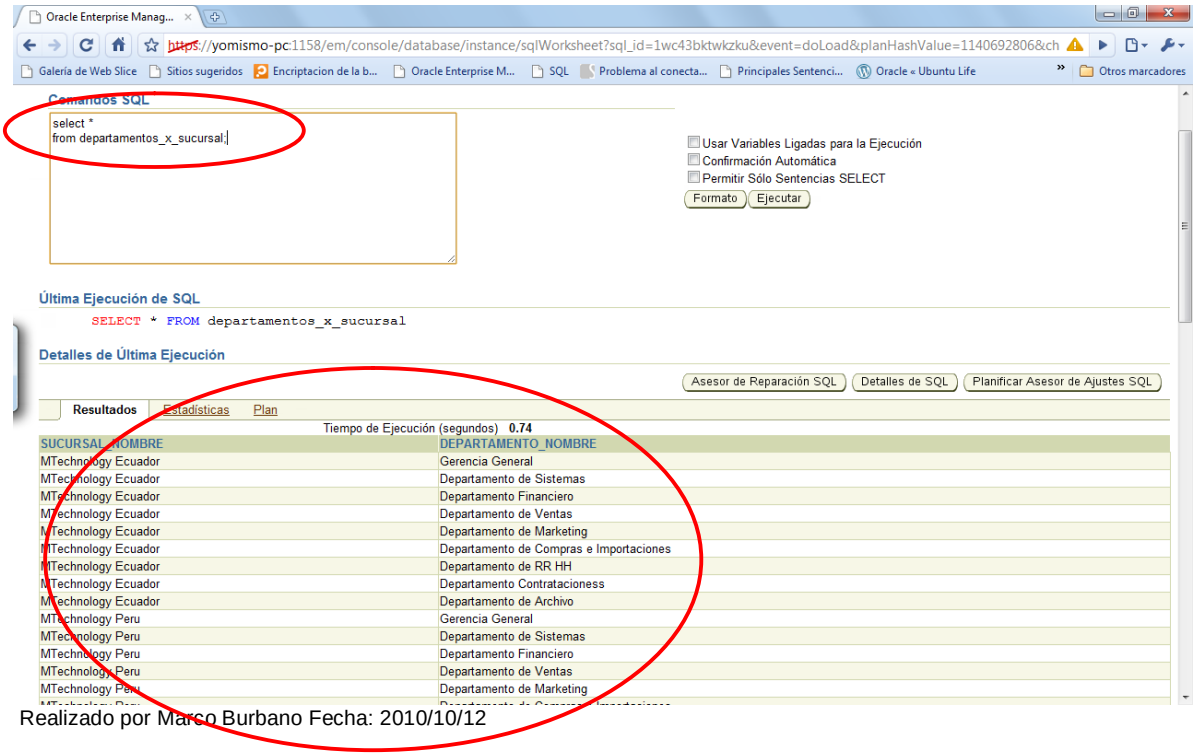
from sucursal, departamento

where departamento.sucursal_id = sucursal.sucursal_id



Realizado por Marco Burbano Fecha: 2010/10/12

Finalmente revisamos si la vista funciona de manera adecuada.



Realizado por Marco Burbano Fecha: 2010/10/12

Adicionalmente podemos verificar en el Enterprise Manager de Oracle si la Vista fue creada y que atributos tiene

Realizado por Marco Burbano Fecha: 2010/10/12

17. Aplicar integridad referencial

Implementación

La integridad de los datos es la propiedad que asegura que la información dada es correcta, al cumplir ciertas aserciones.

Las restricciones de integridad aseguran que la información contenida en una base de datos es correcta, son propiedades de la base de datos que se deben satisfacer en cualquier momento.

Oracle es un sistema de gestión de base de datos (SGBD) relacional que permite la definición de restricciones de integridad dentro del diseño de su base de datos al ser creada. Para incorporar el tratamiento de las restricciones de integridad en el sistema pueden realizarse:

- Añadiendo código adicional para verificar y asegurar que se cumplen las restricciones.
- Declarando las restricciones como parte del esquema de la base de datos.

La definición en la fase de diseño de las restricciones de integridad proporciona mayor número de ventajas, ya que:

- Reduce el costo de desarrollo de software.
- Es más confiable al ser centralizado y uniforme.
- Mantenimiento más fácil.

De acuerdo con la forma de especificación del comando CREATE TABLE dada anteriormente, la cláusula <table_constraint> puede entonces tener las siguientes formas:

- CONSTRAINT <constraint_name> PRIMARY KEY (<column_name>[,<column_name>])
- CONSTRAINT <constraint_name> UNIQUE (<column_name>[,<column_name>])
- CONSTRAINT <constraint_name> FOREIGN KEY (<column_name>[,<column_name>]) REFERENCES <table_name>
- CONSTRAINT <constraint_name> CHECK (<condition>)

Donde:

- <constraint_name> es el nombre con el que se designará al “constraint" en el esquema donde se crea la tabla que lo incluye.
- <column_name> es el nombre de una columna de la tabla en la que se define el “constraint"
- <column_name> es el nombre de una columna de la tabla en la que se define el “constraint"
- <table_name> es el nombre de una tabla definida en el esquema donde existe la tabla que incluye el “constraint".
- <condition> es una expresión lógica de SQL.

18. Utilización de clúster para mejorar la disponibilidad

Implementación[26]

Oracle Real Application Clusters (Oracle RAC), con Oracle Database 11*g* Enterprise Edition, permite ejecutar una sola base de datos en un grupo de servidores y proporciona una tolerancia a fallos, un rendimiento y una capacidad de ampliación inigualables, sin necesidad de cambios de aplicaciones. Los analistas valoran la creciente importancia de Oracle RAC entre el gran número de clientes de todos los sectores que consolidan sus aplicaciones de proceso de transacciones y almacenamiento de datos.

- **Disponibilidad permanente**: Tiempo de actividad ininterrumpido para las aplicaciones de base de datos
- **Capacidad de ampliación bajo demanda**: Amplíe la capacidad con sólo agregar servidores al cluster
- **Flexibilidad**: Virtualice las bases de datos de una instancia con Oracle RAC One Node
- **Menos costes**: Consolide los servidores y reduzca el coste de inactividad
- **Récord mundial de rendimiento**: Funciona a mayor velocidad que el mainframe más rápido

[26]Ver más http://www.oracle.com/technology/global/lad-es/documentation/database.html Consultado el 18 de noviembre de 2010

- **Grid computing**: Oracle RAC es la base de la computación en paralelo (grid computing)

La opción Oracle Real Application Clusters (RAC) soporta la implementación transparente de una sola base de datos a través de un cluster de servidores y brinda tolerancia ante las fallas de hardware o los cortes planificados de servicio. Oracle RAC se ejecuta en cluster y ofrece el máximo nivel de capacidad de Oracle en términos de disponibilidad, escalabilidad e informática de bajo costo. Oracle RAC soporta las aplicaciones comerciales de todo tipo.

Al eliminar el único punto de falla con un solo servidor, Oracle RAC brinda la más alta disponibilidad para sus aplicaciones. Si un nodo en el cluster falla, la Base de Datos Oracle continúa ejecutándose en los nodos restantes. Los nodos individuales pueden dejar de funcionar por motivos de mantenimiento mientras los usuarios de aplicaciones continúan trabajando.

19. Utilización del servicio de un Banco de Datos

Implementación

La utilización del servicio de Banco de Datos no depende directamente del Motor de Base de Datos puesto, que el servicio funciona y lo administra directamente un tercero, quien es el responsable de garantizar la disponibilidad, integridad y seguridad de la información que la organización deposite.

Con lo cual esta opción de poner toda la responsabilidad de la información sobre una tercera persona o institución permite que nuestra empresa se enfoque en su

giro de negocio, así como también tener la certeza que la información está segura y podremos tenerla cuando la necesitemos.

20. Utilización de redundancia

Implementación

Para implementar redundancia controlada en una Base de Datos una de las soluciones que se presentan es la replicación.

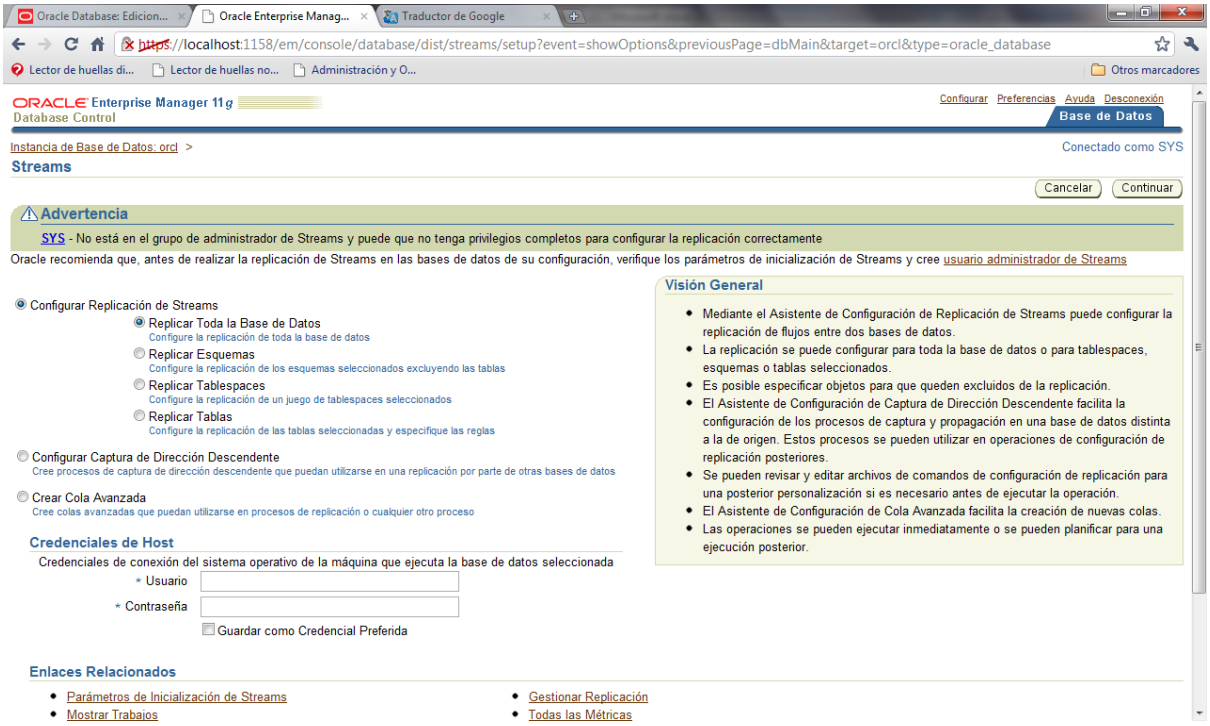
Al utilizar Oracle este motor de base de datos nos brinda las herramientas necesarias y completas para generar una confiable replicación en línea. Esta herramienta es llamada Oracle Streams. La misma que propaga y administra datos, transacciones y eventos en una fuente de datos ya sea dentro de una base de datos, o de una base de datos a otra

Ventajas:

- Permite la propagación y administración de datos, transacciones y eventos, dentro de una BD y/o sobre una BD remota.
- Puede ser utilizado para replicar una BD o una porción de la misma.
- Se utiliza para construir y mantener aplicaciones distribuidas.
- Provee granularidad y control sobre que se va a replicar y como se va a realizar.

Para configurar la ejecución de Oracle Streams podemos ir a la pestaña de “Movimiento de Datos” al Menú de Configuración.

En esta nueva pantalla configuramos la manera y en donde queremos que se replique la base de datos seleccionada.



Realizado por Marco Burbano Fecha: 2010/10/15

CAPITULO 4: ANALISIS COMPARATIVO

Tabla 4: Muestra De Resultados De La Implementación De Las Bases De Datos.

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL	ANÁLISIS
1.	Se debe mantener bloqueados a los usuarios creados automáticamente por la BDD y en el mejor de los casos se los debe Borrar con la finalidad de mantener la seguridad.	Seguridad	En Oracle se mantienen bloqueados hasta que un usuario con permiso los libere y permita su funcionamiento	MySql crea automáticamente el usuario administrador(ROOT), el cual es necesario para el correcto funcionamiento de la BDD	Tanto el motor de base de datos Oracle como MYSQL presentan la posibilidad de administrar usuarios. Así como también nos brindan la posibilidad de gestionar los usuarios creados por la Base de Datos.
2.	Se debe realizar segregación de funciones en el SO y La BDD con la finalidad de no colocar en una sola persona la completa seguridad del ambiente en el que se encuentra la BDD	Seguridad	Se lo realiza evitando una autenticación en la BDD mediante la autenticación del SO y para una mayor seguridad las claves de mayor poder en La BDD y SO deben pertenecer a usuarios diferentes	Se lo realiza evitando una autenticación en la BDD mediante la autenticación del SO y para una mayor seguridad las calves de mayor poder en La BDD y SO deben pertenecer a usuarios diferentes	Es importante realizar una adecuada segregación de funciones puesto que reduce el riesgo accidental o deliberado del mal uso del sistema. Cuando sea difícil de segregar, deben ser considerados otros controles como el monitoreo de actividades, pistas de auditoría y supervisión de los administradores y tener cuidado que

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL	ANÁLISIS
					ninguna persona pueda realizar actividades fraudulentas en áreas de responsabilidad única sin que pueda ser detectada.
3.	Mediante aplicación o si la BDD lo permite pedir el cambio de contraseña obligatorio después de la creación de un usuario	Seguridad	Oracle lo ha implementado automáticamente incluso se puede definir una periodicidad para solicitar cambios de contraseñas	Se lo puede controlar mediante una aplicación que funcione con la BDD, así como también automatizar la petición de cambio de contraseña.	Oracle definitivamente es una de las bases de datos más completas del mercado, ya que nos permite solicitar el cambio de contraseña después de la creación del usuario, mientras que MySQL este punto debemos complementarlo con líneas de código de la aplicación.

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL	ANÁLISIS
4.	Restringir el acceso de usuarios autorizados a la BDD	Seguridad	Se debe definir un método de acceso a la BDD el cual puede ser mediante el uso de un usuario y contraseña autorizados o cualquier otro método eficaz	Se debe definir un método de acceso a la BDD el cual puede ser mediante el uso de un usuario y contraseña autorizados o cualquier otro método eficaz	En este punto Oracle lleva una completa diferencia en la administración de las contraseñas proporcionando no solo como medio de validación un usuario y password sino también la posibilidad de acceso mediante el sistema operativo sobre el cual se ha instalado la BDD, o también se puede delegar este trabajo a un tercer software de Oracle.
5.	Para la ejecución de scripts o líneas de comando estas deben tener previa autorización más aún si los escripts tienen incluidas contraseñas de usuarios con poder en la BDD	Seguridad	Definir que scripts y con la autorización de quien dentro del departamento de TI son los autorizados para ser ejecutados en la BDD	Definir que scripts y con la autorización de quien dentro del departamento de TI son los autorizados para ser ejecutados en la BDD	Para la implementación de este punto en cualquiera que sea la base de datos, se debe definir políticas acordes a la situación y desempeño de la organización. Pero para la restricción de acceso a líneas de comando es necesario limitar el número de interfaces instaladas en toda la organización y que las que han sido instaladas pertenezcan a usuarios autorizados generalmente del departamento

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL	ANÁLISIS
					técnico.
6.	Limitar el número de sesiones concurrentes por cada usuario de preferencia permitir una sola sesión por cada usuario	Seguridad	Oracle permite esta restricción	MySql permite esta restricción	La implementación de esta práctica se la puede realizar tanto en MySQL como en Oracle mediante líneas de código como mediante sus respectivas Herramientas gráficas, poniendo a cada uno de los usuarios el mínimo numero de sesiones correspondientes a su rol.
7.	La posibilidad de usar “GRANT OPTION” o “REVOKE” debe ser permitido solo a personal autorizado	Seguridad	Oracle permite esta restricción	MySql permite esta restricción	Tanto Oracle como MySQL permiten la posibilidad de administrar tanto el REVOQUE como el GRAN OPTION de manera que solo personal autorizado tenga estos privilegios.
8.	Denegar la posibilidad de conectarse remotamente a la BDD más aún si los usuarios tienen permisos para modificar la información	Seguridad	Oracle permite esta restricción	MySql permite esta restricción	Oracle y MySQL nos permiten la limitar la conexión remota hacia la base de datos. Utilizando sus respectivas herramientas o

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL	ANÁLISIS
	sensitiva				funcionalidades
9.	La información sensitiva almacenada en la Base de Datos debería ser encriptada. El no tener la información primordial encriptada aumenta el riesgo de accesos no autorizados.	Seguridad	Oracle permite esta restricción	MySql permite esta restricción	La información sensitiva de la base de datos como son las contraseñas de los usuarios son codificadas con sus respectivas funciones de encripción. Pero es importante mencionar que Oracle es compatible con varios paquetes de encripción que fortalece la seguridad de la base de datos.
10.	Activar los registros de auditoría con la finalidad de llevar un control sobre las actividades realizadas en la BDD	Seguridad	Oracle permite esta restricción	Se lo debería implementar mediante un Trigger o un aplicativo	MySQL no tiene implementado una tabla de registros de auditoría, la cual contenga todas las actividades realizadas que puedan poner en peligro la seguridad o disponibilidad de la BDD. Por otro lado Oracle ofrece una gran versatilidad en la generación de pistas de auditoría puesto que aparte de tener esta funcionalidad incluida en la herramienta, esta es configurable de manera que satisfaga

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL	ANÁLISIS
					las necesidades de cada una de las organizaciones.
11.	El administrador de seguridad (no el DBA) debe supervisar la configuración de auditoría de base de datos. Sólo el administrador de seguridad debe tener acceso a los registros de auditoría.	Seguridad	Se lo implementa mediante roles y permisos	Se lo implementa mediante roles y permisos	Oracle y MySQL tienen la posibilidad de implementar roles a los cuales asignar diferentes permisos para poder realizar las actividades permitidas a cada usuario. Es importante mencionar que MySQL no tiene las características para aplicar roles pero si se puede restringir a cada usuario sus diferentes permisos,
12.	Asegurar que los controles de Respaldo y Restauración de la Base de Datos garantiza la disponibilidad de los datos los cuales se pueden recuperar por completo	Seguridad/ Disponibilidad	Oracle permite ejecutar respaldos la revisión va por cuenta del departamento de IT	MySql permite ejecutar respaldos la revisión va por cuenta del departamento de IT	La implementación de una política de respaldos es responsabilidad de los responsables de dirigir el departamento de IT, mientras la obtención de los backups va por cuneta de las Bases de Datos y tanto Oracle como MySQL permiten la ejecución de backups periódicos de manera manual.

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL	ANÁLISIS
13.	Realizar una correcta asignación de privilegios a cada uno de los usuarios	Seguridad/ Integridad	Implementar procedimientos rigurosos para la asignación de perfiles y roles en la BDD	Implementar procedimientos rigurosos para la asignación de perfiles y roles en la BDD	Generar una correcta asignación de privilegios genera un ambiente de control adecuado sobre las actividades que se realizan en la organización. Es por esto que Oracle mediante roles nos facilita asignación de privilegios, pero MySQL si bien no tiene la opción de roles si nos permite restringir permisos a los usarios y otorgar solo las funciones necesarias a cada uno de los usuarios.
14.	Todos los identificadores de usuario debe ser único e identificar un único usuario y debe respetar la norma de denominación corporativa.	Seguridad	Definir políticas para el acceso a la BDD	Definir políticas para el acceso a la BDD	Este aspecto es definido de manera automática por la base de datos, puesto que controlan el acceso a la base de datos mediante un usuario y contraseña, en donde el usuario debe ser único y solo con contraseña valida se puede ingresar. Pero para tener un control adecuado se debe crear un procedimiento formal de creación de cuantas de usuario, ya que la creación deliberada de usuarios genera un riesgo de

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL	ANÁLISIS
					seguridad e integridad a la base de datos.
15.	Restringir el uso de líneas de comando a usuarios autorizados	Seguridad	Configuración del ambiente de control en el departamento de TI	Configuración del ambiente de control en el departamento de TI	Este punto depende completamente de la organización y control que se realiza desde los responsables del departamento de tecnología. Evitando el acceso a líneas de comando desde interfaces no autorizadas.
16.	Aplicación de vistas con la finalidad de restringir la información a usuarios finales	Integridad	Oracle permite la realización de Vistas	MySql permite la realización de Vistas	La utilización de vistas es una de las practicas mas básicas pero eficaces para mantener la integridad y seguridad de las bases de datos ya que con la implementación de vistas se puede mostrar solamente lo necesario y ocultar lo que el usuario final no debe conocer. Tanto Oracle como MySQL permiten implementar vistas.

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL	ANÁLISIS
17.	Aplicar integridad referencial	Integridad	Implementado en Oracle	Implementado en MySql con Tablas INNODB	La integridad referencial ha sido aplicado en las tablas INNODB de MySQL, mientras la integridad referencial es implementada en Oracle con lo cual se ha posicionada como uno de los más grandes motores de BD.
18.	Utilización de clúster para mejorar la disponibilidad	Disponibilidad	Oracle permite la realización de clúster mediante una configuración en el Motor de BDD	MySQL no tiene esta función integrada, pero existe la posibilidad de configurar clúster de manera manual. Con la utilización de un paquete adicional de MySQL	Tanto Oracle como MySQL permite la implementación de Clusters lo cual incrementa la disponibilidad de la Base de Datos.
19.	Utilización del servicio de un Banco de Datos	Disponibilidad	No depende de la BDD local	No depende de la BDD local	No depende de la BDD local, esto debe ser implementado por una tercera organización que se responsabilice de la información de la empresa,

No.	DESCRIPCIÓN	CLASIFICACIÓN	IMPLEMENTACIÓN ORACLE	IMPLEMENTACIÓN MYSQL	ANÁLISIS
20.	Utilización de redundancia	Disponibilidad	Oracle permite generar redundancia	Mysql permite realizar replicación para mantener redundancia controlada.	Tanto Oracle como MySQL permite realizar replicación con la finalidad de mantener una redundancia controlada. Cada uno de estos motores de bases de datos mediante sus herramientas permiten la configuración de replicación.

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- La implementación de las prácticas definidas en este trabajo constituye un enfoque que permiten obtener resultados satisfactorios de seguridad, integridad y disponibilidad en la implementación de bases de batos

- Prácticas de seguridad, integridad y disponibilidad para bases de batos se encuentran muy relacionadas, por lo que es necesario vincular un plan de seguridad de la información que abarque estas prácticas de manera total. Puesto que si solo nos enfocamos en una sola actividad, la efectividad del plan no presenta los mismos resultados

- Para que se mantenga niveles adecuados de seguridad, integridad y disponibilidad en las configuraciones de bases de datos, no solo depende de la parte tecnológica y de la configuración que se implemente en los motores de bases de datos. Una gran parte de la responsabilidad para mantener la seguridad, integridad y disponibilidad recae sobre las actividades que realizan los integrantes

de la organización como son la implementación de manuales, políticas y procesos que complemente la configuración tecnológica.

- Una de las ventajas de utilizar estándares radica en el hecho de utilizar experiencias de otras organizaciones en beneficio propio, lo que ahorra tiempo y recursos. Por otro lado existen regulaciones que recomiendan a las organizaciones basar sus procesos en mejores prácticas de la industria.

- La seguridad, integridad y disponibilidad de la información sensible pueden llegar a ser esenciales para mantener los niveles de competitividad y sobre todo continuidad del negocio.

- El nivel de seguridad alcanzado por los medios técnicos demuestra ser limitado e insuficiente por sí mismo, en la gestión efectiva de la seguridad debe tomar parte activa toda la organización.

- Garantizar un nivel de protección total es imposible incluso en el caso de disponer de un presupuesto ilimitado. El propósito de esta guía es gestionar los riesgos de la seguridad, integridad y disponibilidad que se pueden presentar en las bases de datos.

- Al hablar de mejores prácticas no se puede dejar de lado el desarrollo de un plan de contingencia, en donde de manera obligada es

necesario tomar en cuenta la seguridad de la información e incluir medidas de seguridad para garantizar el resguardo de la información, puesto que en situaciones de emergencia el principal activo que quizá nunca se pueda recuperar es la información.

5.2 Recomendaciones

- Se recomienda el estudio a través de organismos especializados y el análisis de herramientas disponibles como son las normas ISO, la gestión de IT como COBIT e ITIL para mejorar los servicios Tecnológicos.

- Se recomienda la implementación de las actividades descritas en el presente proyecto, puesto que con la aplicación de las actividades detalladas se puede administrar de una mejor manera el ambiente de control para la administración de las bases de datos.

- Se recomienda realizar un análisis completo del área de sistemas de las organizaciones previo a la implementación de las mejores prácticas, identificando las políticas, estándares y procedimientos que se encuentran documentados o simplemente en el conocimiento y experiencia de los integrantes de este departamento. Puesto que es necesario conocer y evaluar el ambiente de control y así determinar la

situación y correcciones necesarias para implementar las correspondientes soluciones.

- También es importante recomendar que la organización previo a la implementación de las mejores prácticas, realicen el levantamiento o inventario de los equipos y sistemas que posee la organización. Ya que se necesita conocer los activos tecnológicos que se tienen y sobre los cuales vamos a implementar las practicas de integridad, disponibilidad y seguridad.

- Se recomienda que se tomen en cuenta todos los consejos aquí discutidos, pero es importante mencionar que cada una de las empresas son las llamadas a tomar la decisión de cuál es el motor de base de datos que más conviene al giro del negocio, y realizar un análisis de cuáles son las practicas viables de implementar y cuáles no lo son.

6. REFERENCIAS BIBLIOGRÁFICAS

Best, Tom, y M J Bilings. *Oracle Data Base 10g: Administration Workshop I.* Edición 3.0, Noviembre 2005.

Greenberg, Nancy, y Prya Nathan. *Introduction to Oracle 9i: SQL.* 2001.

Guapás, L Miguel A. *Manual de Entrenamiento y Referencia en las mejores Prácticas de Itil V3.* 2008.

Ramos, M J, y F Montero. *CEO Sistemas gestores de Bases de Datos.* McGraw-Hill.

Codd, Edgar. *Departamento de Ciencias de la Computación .* http://www.dcc.uchile.cl/~rbaeza/inf/codd.html.

Consejo Superior de Administración Electrónica. http://www.csae.map.es/csi/silice/Sgbdat6.html.

Departamento de Informática. 09 de Junio de 2010. http://www.infor.uva.es/~jvegas/cursos/bd/oraseg/oraseg.html#4.1.

MSDN. 2010. http://msdn.microsoft.com/es-es/library/ms190174.aspx.

MySQL. 2010. http://dev.mysql.com/doc/refman/5.0/es/user-names.html.

MySQL. 2010. http://dev.mysql.com/doc/refman/5.0/es/ndbcluster.html.

Oracle. *Oracle.com.* http://download-west.oracle.com/docs/cd/B14117_01/appdev.101/b10802/d_obtool.htm#1002213.

—. *Oracle.com.* http://www.oracle.com/technology/global/lad-es/documentation/collaterals/Oracle%20Database%2011g%20High%20Availability_cast_.pdf.

—. *Oracle.com.* http://www.oracle.com/technology/global/lad-es/documentation/database.html.

University, Estanford. http://www.stanford.edu/dept/itss/docs/oracle/10g/server.101/b10759/statements_5005.htm.

Web Taller. 5 de 11 de 2010. http://www.webtaller.com/construccion/lenguajes/mysql/lecciones/tipos_tablas_usadas_mysql.php.

Aula CLic. Julio de 2000. http://www.aulaclic.es/sql/b_8_1_1.htm.

Wikipedia. 26 de Octubre de 2010. http://es.wikipedia.org/wiki/Información (último acceso: Abril de 2010).

Wikipedia. 4 de Octubre de 2010. http://es.wikipedia.org/wiki/Base_de_datos.

Wikipedia. Mayo de 2010. http://es.wikipedia.org/wiki/Administrador_de_base_de_datos.

Wikipedia. 22 de Mayo de 2010. http://es.wikipedia.org/wiki/Cluster_(informática).

7. ANEXOS

7.1 Glosario

BDD	Base de Datos.
SO	Sistema Operativo
DBA	Date Base Administrator o Administrador de la Base de Datos
ID	Identificador
SQL	Structured Query Language o Lenguaje estructurado de Consultas.
DBASE	El primer Sistema de gestión de base de datos usado ampliamente para microcomputadoras, publicado por Ashton-Tate para CP/M, y más tarde para Apple.
Cubos OLAP	*On Line Analytical Processing* o Procesamiento Analítico en Línea.
Hard-Code	Mala práctica en el desarrollo e implementación de sistemas, en el cual se insertan datos en el código fuente.
Scripts	Archivo de órdenes, por lo general se almacena en un archivo plano

Tupla	Arreglo horizontal de una tabla (registro)
Cluster	Conjunto de ordenadores construidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una único ordenador.
Backup	Copia de Seguridad